

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



Sistema de Aprovisionamiento para la Red DOCSIS de Telecable Internacional de Cayo Santa María

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

Autor:
Deivis González González.

Tutores:
Lic. Yaima Toledo Guerra. UCF Carlos Rafael Rodríguez.
Ing. Alejandro León Rodríguez. UCF Carlos Rafael Rodríguez

Cienfuegos, Cuba
Curso 2012 - 2013

DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo de diploma titulado “Sistema de Aprovechamiento para la Red DOCSIS de Telecable Internacional de Cayo Santa María” y por este medio reconozco al Departamento de Informática de la Facultad de Ingeniería de la Universidad de Cienfuegos “Carlos Rafael Rodríguez” con todos los derechos patrimoniales del mismo. Para que así conste firmamos la presente a los ____ días del mes de_____ del 2013.

Autor: Deivis González González

Tutor: Lic. Yaima Toledo Guerra

Tutor: Ing. Alejandro León Rodríguez

PENSAMIENTO

"La seguridad no es un producto, es un proceso", y "la seguridad de la información es un asunto de personas, procesos y tecnología"

Schneier

DEDICATORIA

Se la dedico a toda mi familia y mis amistades.

AGRADECIMIENTOS

*Gracias a mi mamá, a mi papá, a mi mis abuelos y toda mi familia
por apoyarme siempre, especialmente en los momentos
en que más lo necesité.*

*Gracias a mis tutores Alejandro y Yaima por compartir conmigo todos sus
conocimientos, su ayuda y su tiempo.*

*Gracias también a mi novia Dailit que me ha apoyado en todo
este camino de la tesis*

*Gracias a mi mejor amigo Yoel que me apoyó en las buenas
y en las malas y nunca me falló.*

*Gracias a todos mis compañeros de estudio que han estado conmigo
siempre en las buenas y en las malas*

*Gracias a todos los profesores de la facultad que han puesto de su parte
para que podamos aprender cada día más.*

A todos muchísimas gracias.

RESUMEN

La presente investigación lleva por título “Sistema de Aprovisionamiento para la Red DOCSIS de Telecable Internacional Cayo Santa María” como alternativa de solución a las deficiencias del aprovisionamiento de la red DOCSIS de dicha entidad. Tiene como objetivo el aprovisionar dinámicamente y de fácil configuración esta red DOCSIS, a través de la conformación de archivos de configuración para abonados y de la configuración de los servicios DHCP y TFTP, permitiendo el futuro desarrollo de servicios como internet, video a demanda entre otros, con el propósito de mejorar las actividades y ampliar la cartera de ofertas en este polo turístico.

En el documento se describen los elementos que conforman el análisis, diseño e implementación del sistema propuesto, siguiendo lo establecido por SCRUM y utilizando el Lenguaje Unificado de Modelado (UML). Se utilizó además PostgreSQL como sistema gestor de bases de datos y Java como lenguaje de programación.

ABSTRACT

This investigation, entitled “Provisioning System for Telecable Internacional Santa María Keys’ DOCSIS Network”, pretends to solve the provisioning deficiencies of its company DOCSIS Network. It has the goal of dynamically provide the network through the creation or edition of configuration files, and through the configuration of DHCP and TFTP services, making possible to offer services like Internet, video on demand, among others, giving a wider portfolio to this Cuban tourist center.

In this document the elements about designing, analysis and implementation of the proposed system are given, following SCRUM guidelines and using the unified modeling language (UML). PostgreSQL was also used as database manager and Java as the programming language.

ÍNDICE

INTRODUCCIÓN	1
CAPÍTULO 1. MARCO TEÓRICO Y CONCEPTUAL	5
1.1 Introducción.....	5
1.2 Las Redes de Televisión y Datos.....	5
1.2.1 Arquitectura de las redes de Televisión por Cable.....	5
1.2.2 Redes Híbridas de Fibra y Coaxial (HFC).....	6
1.2.3 Transmisión de datos de redes de HFC.	8
1.2.4 DOCSIS (Data Over Cable Service Interface Specification).....	8
1.2.5 Arquitectura.	9
1.2.6 Características y Funcionamiento General.	10
1.2.7 Operación básica del protocolo.	11
1.3 Aprovisionamiento en redes DOCSIS.	13
1.3.1 Protocolo de Configuración Dinámica de Host.	13
1.3.2 Protocolo de Transferencia de Ficheros Trivial.....	14
1.4 Flujo actual de los procesos y análisis de la ejecución.	14
1.5 Análisis de los Software existentes.	15
1.6 Metodología de Desarrollo de Software.	16
1.6.1 Metodologías Tradicionales.	16
1.6.2 Metodologías Ágiles.....	17
1.7 Lenguajes utilizados.....	19
1.7.1 Lenguaje de Modelado.	19
1.7.2 Lenguaje de Programación.....	20
1.7.3 Lenguaje SQL.....	25
1.8 Sistemas Gestores de Base de Datos.	25
1.9 Herramientas Utilizadas	26
1.8.1 Visual Paradigm.....	27
1.9 Conclusiones del Capítulo.	28
CAPÍTULO 2. DESCRIPCIÓN Y CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA.....	29
2.1 Introducción al Capítulo.	29
2.2 Pila del Producto.....	29
2.3 Requerimientos del Sistema.	32

2.3.1 Requisitos No Funcionales.	32
2.4 Planeación de los Sprint.....	36
2.4.1 Listado de los Sprint.	36
2.4.2 Técnicas de estimación de Sprint.	37
2.4.3 Descripción de los Sprint.	38
2.5 Diagrama de Caso de uso.....	47
2.5.1 Descripción de los casos de uso.....	48
2.6 Diagramas.....	48
2.7 Principios del diseño del sistema.	50
2.7.1 Diseño de la interfaz de entrada, salida y menús del sistema.	51
2.7.2 Tratamiento de excepciones.....	51
2.7.3 Estándares de Codificación.	51
2.8 Conclusiones del Capítulo.	52
CAPÍTULO 3. ESTUDIO DE FACTIBILIDAD Y VALIDACIÓN DEL SISTEMA.	53
3.1 Introducción.....	53
3.2 Factibilidad.....	53
3.2.1 Cálculo de Puntos por Casos de Uso sin Ajustar.....	53
3.2.2 Factor de Peso de los Actores sin ajustar (UAW).....	54
3.2.3 Factor de Peso de los Casos de Uso sin Ajustar.....	55
3.2.4 Cálculo de Puntos de Casos de Uso ajustados.	56
3.2.5 Factor de complejidad técnica (TCF).	57
3.2.6 Factor ambiente (EF).	58
3.2.7 Estimación del esfuerzo.....	59
3.2.8 Cálculo de costos:.....	61
3.2.9 Beneficios tangibles e intangibles.	61
3.2.10 Análisis de los costos y beneficios.....	62
3.3 Validación de la solución propuesta.....	62
3.4 Conclusiones del Capítulo.	64
CONCLUSIONES.....	65
RECOMENDACIONES	66
REFERENCIAS BIBLIOGRÁFICAS	67
BIBLIOGRAFÍA	68
ANEXOS	71

ÍNDICE DE FIGURAS

Figura 1: Arquitectura básica de una red de Televisión por Cable (CATV).....	6
Figura 2: Arquitectura de una red de cable bidireccional.	6
Figura 3: Topología de una red HFC.....	7
Figura 4: Arquitectura de una red HFC con DOCSIS.....	9
Figura 5: Componentes básicos de una Red DOCSIS.....	12
Figura 6: Diagrama de Casos de Usos	48
Figura 7: Modelo lógico de la base de datos Esquema Usuarios.....	49
Figura 8: Modelo lógico de la base de datos Esquema Sistema.....	49
Figura 9: Modelo físico de la base de datos Esquema Usuarios.....	50
Figura 10: Modelo físico de la base de datos Esquema Sistema.....	50
Figura 11: Comparación del tiempo promedio de los procesos antes y después del sistema.	63

ÍNDICE DE TABLAS

Tabla 1: Velocidades para los canales de bajada de 6 MHz.	11
Tabla 2: Pila del Producto	32
Tabla 3: Listado de los Sprint.....	37
Tabla 4: Pila del Sprint 1.	38
Tabla 5: Estimación de Historia del Sprint 1.....	39
Tabla 6: Historias divididas en tareas Sprint 1.	40
Tabla 7: Pila del Sprint 2.	41
Tabla 8: Cálculo de la velocidad estimada para el Sprint 2.....	41
Tabla 9: Historias divididas en tareas Sprint 2	42
Tabla 10: Pila del Sprint 3.	42
Tabla 11: Cálculo de la velocidad estimada para el Sprint 3.....	42
Tabla 12: Historias divididas en tareas Sprint 3	44
Tabla 13: Pila del Sprint 4.	44
Tabla 14: Cálculo de la velocidad estimada para el Sprint 4.....	45
Tabla 15: Historias divididas en tareas Sprint 4	46
Tabla 16: Pila del Sprint 5.	46
Tabla 17: Cálculo de la velocidad estimada para el Sprint 4.....	46
Tabla 18: Historias divididas en tareas Sprint 5	47
Tabla 19: Descripción de los Casos de Uso.....	48
Tabla 20: Criterios del factor de peso de los actores sin ajustar.	54
Tabla 21: Clasificación de los actores del sistema.	54
Tabla 22: Criterios del factor de peso de los casos de uso sin ajustar.....	55
Tabla 23: Clasificador de los casos de uso del sistema.	56
Tabla 24: Significado y pesos de los TCF.	58
Tabla 25: Significado y peso de las habilidades del grupo.....	59
Tabla 26: Criterios de distribución de esfuerzos.....	61

INTRODUCCIÓN

La necesidad de ver una imagen televisiva clara, sin ruido ni interferencias apreciables, dio lugar a la creación de la industria de la televisión por cable cuando la forma más común de transmisión de televisión era mediante la utilización del espectro electromagnético por aire, que en Estados Unidos y la mayoría de los países, se remonta a la década de 1940. 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.[1]

Posteriormente, al aumentar la competencia de las empresas de comunicaciones por brindar el servicio de Internet, a las empresas de TV por cable, se les ocurrió que por ese mismo cable que ya llegaba a tantas personas, podrían dar el servicio de internet también. Así surge en la década de los 90' las redes DOCSIS, que por sus siglas significa Especificación de la Interfaz del Servicio de Datos sobre Cable y que no es más que las antiguas redes de cable con otro equipamiento en sus extremos que permiten además de la televisión, la transmisión de datos como internet, audio y vídeo en tiempo real, todo empleando el mismo cable ya instalado.[2]

La industria del cable, como se le conoce, partiendo de las necesidades de los clientes, muchas veces impuestas, fue evolucionando hasta llegar a donde está hoy, impulsada por la constante competencia de otros servicios que también se expandieron rápidamente como los servicios de línea de suscriptor digital (DSL).[3]

Como resultado de un proceso de unificación de métodos y conceptos se cuenta hoy con 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 cable. Muchos operadores de televisión por cable lo

emplean para proporcionar acceso a Internet sobre una infraestructura HFC (red híbrida de fibra óptica y coaxial) existente.[4]

La forma en que se realizan tanto las funciones de control y operación de los módems de cable como las de acuerdos de nivel de servicio, son las que caracterizan al proceso de Aprovisionamiento de Red, conociéndose a la plataforma o aplicación cuya función tributa directamente a este proceso como Sistema de Aprovisionamiento.

La realidad de hoy muestra un mercado de software de gestión para equipamiento DOCSIS muy reducido, permitiendo altos precios a los pocos que ofrecen el producto. En Cuba, se cuenta con el equipamiento básico para brindar los servicios DOCSIS en la infraestructura hotelera de Cayo Santa María, pero la carencia de un Sistema de Aprovisionamiento hace muy difícil su implementación. 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.

De lo anterior expuesto se deduce el siguiente **problema de investigación**: ¿Cómo implementar un Sistema de Aprovisionamiento para la red DOCSIS de Telecable Internacional Cayo Santa María que permita, en principio, comenzar a brindar los servicios de transmisión de datos?

Se plantea como **idea a defender**: El desarrollo de un sistema informático que aprovisione los dispositivos de esta red DOCSIS, facilitará el proceso de aprovisionamiento de la red permitiendo un servicio de datos basado en acuerdos de nivel de servicio de manera dinámica.

Como consecuencia se define el siguiente **objetivo general** para esta investigación: Elaborar un Sistema de Aprovisionamiento para la red DOCSIS de Telecable Internacional Cayo Santa María que responda a las necesidades del usuario.

Y los siguientes objetivos específicos:

- Estudiar del estándar DOCSIS y la descripción de sus principales características.
- Analizar los protocolos de administración de redes y sus implementaciones por los diferentes lenguajes de programación.
- Implementar el sistema de aprovisionamiento.
- Comprobar de los resultados en un entorno real.

Los anteriores objetivos se cumplen a través del término de las siguientes tareas:

- Recopilación de la información necesaria para el análisis de los temas relacionados con el estudio de las redes DOCSIS.
- Elaboración de un Sistema de Aprovisionamiento utilizando un lenguaje de programación.
- Realización de pruebas en el terreno para ver los resultados del diseño y su posible aplicación práctica.
- Conformación de un interfaz usuario-sistema.
- Evaluación de la efectividad de la propuesta en el entorno de red

El presente informe de resultados está estructurado por Resumen, Introducción, Capítulos I, II y III; Conclusiones, Recomendaciones y Anexos correspondientes.

En el Capítulo I. Fundamentación Teórica: se fundamentan y analizan críticamente los principales conceptos vinculados al campo de acción; así como los antecedentes y sistemas informáticos vinculados al aprovisionamiento de redes DOCSIS. Se documenta el estudio realizado sobre las tendencias, tecnologías y herramientas actuales, en estas se fundamentan la selección de las utilizadas en la solución propuesta.

Capítulo II. Planificación y control para el desarrollo del Software: se fundamenta como guía para el desarrollo del software la metodología Scrum. Se completan la pila del producto y los requerimientos no funcionales, se

realiza la planeación de cada sprint, así como las técnicas de estimación de estos.

Capítulo III: En este capítulo se tratan los diferentes aspectos relacionados al estudio de la factibilidad del sistema. Se estiman el esfuerzo humano y el tiempo de desarrollo que se requieren para la elaboración del mismo. Se analizan los costos y los beneficios para concluir si es factible o no desarrollar del sistema. Se realiza además la validación del sistema mediante una Prueba T, para medir las diferencias de tiempo de los diferentes procesos antes y después de realizado el sistema informático.

CAPÍTULO 1. MARCO TEÓRICO Y CONCEPTUAL.

1.1 Introducción.

En este capítulo se abordan los diferentes aspectos teóricos relacionados con los sistemas de Aprovechamiento en redes DOCSIS. Este capítulo resume toda una etapa de búsqueda y análisis de información referente al objeto de estudio, sistemas existentes, tecnologías y tendencias actuales, se justifica la metodología utilizada, así como cada una de las herramientas utilizadas en el desarrollo del sistema informático.

1.2 Las Redes de Televisión y Datos.

La evolución de las tecnologías de la información y las comunicaciones ha venido definiendo una tendencia a la integración de los servicios en la que el mundo IP (internet) ha ido penetrando en casi todas las variantes de redes de comunicaciones.

A esto no han estado ajenas las redes de televisión, específicamente las de televisión por cable. Estas y la forma en la que han incorporado nuevos servicios, especialmente el de datos, serán tratadas a continuación.

1.2.1 Arquitectura de las redes de Televisión por Cable.

El lugar donde se encuentran los equipos de recepción de señales, generadores de programación local, donde se procesa y se conforma el paquete de canales así como donde se gestionan los servicios adicionales que se ofrecen, se denomina cabecera.[5]

Una red de cable se compone además por los tendidos de conductores y dispositivos de distribución y la red de abonado, como se muestra en la Figura 1.1.

CAPÍTULO I. MARCO TEÓRICO Y CONCEPTUAL.

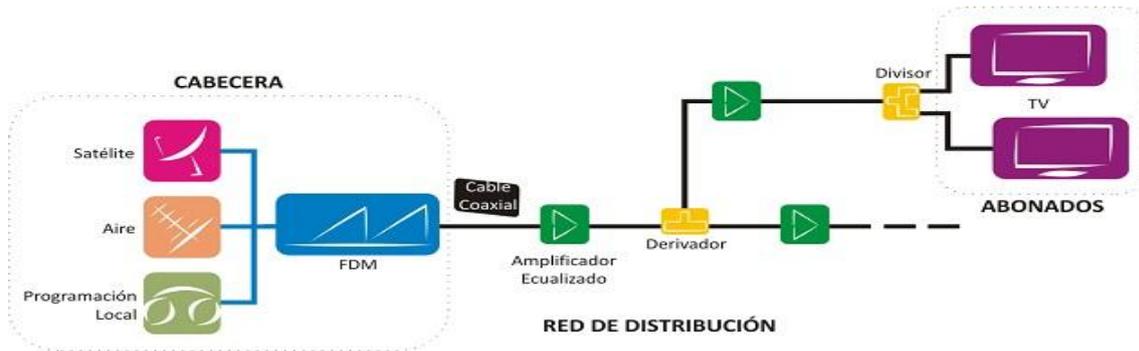


Figura 1: Arquitectura básica de una red de Televisión por Cable (CATV).

Los sistemas de cable fueron concebidos para difusión de televisión, requiriendo solamente una transmisión unidireccional. Sin embargo servicios como Internet y Pay-per-View necesitan una comunicación en dos sentidos, razón por la cual los operadores han tenido que actualizar sus redes con las nuevas arquitecturas. Así, se separa una banda para la comunicación 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. La comunicación se establece entre equipos terminales en la cabecera y en el extremo del suscriptor, [2]de manera similar a la mostrada en la figura 1.2.

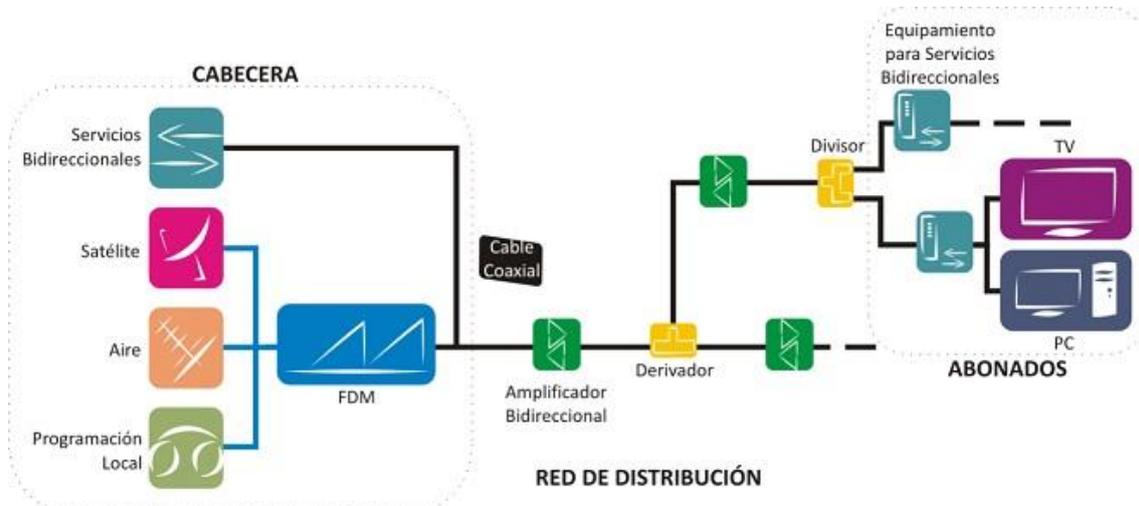


Figura 2: Arquitectura de una red de cable bidireccional.

1.2.2 Redes Híbridas de Fibra y Coaxial (HFC).

La transmisión bidireccional en redes de sólo coaxial experimentó grandes pérdidas de señal y eran muy propensas al ruido en las conexiones,

CAPÍTULO I. MARCO TEÓRICO Y CONCEPTUAL.

especialmente en el canal de retorno. En la década de los noventa aparecen las Redes Híbridas de Fibra y Coaxial (HFC), como una modificación en el sistema de transmisión para aumentar el alcance, reducir el ruido y las distorsiones, incrementar la fiabilidad, disminuir gastos de mantenimiento y proveer medios económicos para actualizar las redes existentes, creando además una vía para interconectar las cabeceras.[6]

Las redes HFC aprovechan significativamente la menor atenuación que sufren las señales al ser transportadas sobre sistemas de fibra óptica[6]. Las señales eléctricas se insertan en un convertidor óptico-eléctrico que las entrega para el transporte ópticamente moduladas. Usualmente, la fibra óptica conecta la cabecera con nodos ópticos distribuidos en una región, donde se recuperan nuevamente las señales eléctricas y se distribuyen por cable coaxial hasta los consumidores. Esto permite aislar el ruido hasta los nodos y disminuir las pérdidas de señal del sistema, permitiendo construir redes más grandes (Tooley and Bowman, 2010), que ofrecen un escenario ideal para la transmisión bidireccional.[7] Una red HFC se conforma básicamente como lo muestra la figura 1.3.

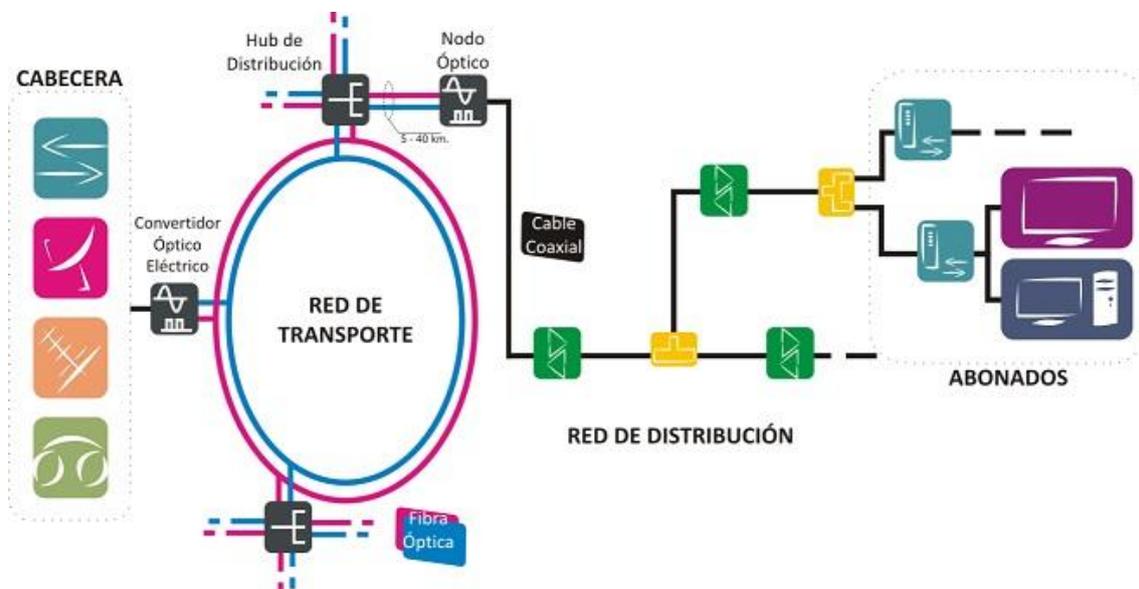


Figura 3: Topología de una red HFC.

Las redes HFC son únicas en cuanto a:

- Soportan la operación en dos sentidos sobre el mismo cable.

CAPÍTULO I. MARCO TEÓRICO Y CONCEPTUAL.

- Son asimétricas. Debido a su origen en sistemas de difusión de televisión se cuenta con más ancho de banda para downstream que para upstream.
- Emplean transporte común para múltiples servicios, MPEG2, que carga tanto paquetes de datos como señales de televisión digital.[7]

1.2.3 Transmisión de datos de redes de HFC.

Durante los años 90, la industria del cable desarrolló un gran número de esquemas para soportar la transmisión de datos en redes CATV/HFC, por lo que varios estándares emergieron en calidad de competidores.

- **IEEE 802.14:** En 1994 este grupo de trabajo desarrolló una capa MAC para soportar ATM e IP sobre redes HFC. El canal de subida era TDMA con un tamaño de ranura de 8 bytes.
- **Multimedia Cable Network System's (MCNS) DOCSIS:** En respuesta a la competencia de DSL, varios Operadores de Servicios Múltiples (MSOs) crearon el MCNS 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 *Digital Audio Visual Council* (DAVIC) produjo el DAVIC 1.2 y el muy similar *Digital Video Broadcast Return Channel for Cable* (DVB-RCC), ambos estándares para cable módems, que definieron las capas Física (PHY) y MAC para comunicaciones bidireccionales sobre redes CATV/HFC. Posteriormente la industria europea se movió hacia EuroDOCSIS, lógicamente por motivos económicos.[1]

1.2.4 DOCSIS (Data Over Cable Service Interface Specification).

Finalmente DOCSIS fue el estándar que se consolidó y fue adoptado por los grandes fabricantes de equipos de comunicaciones como Cisco y Broadcom.

DOCSIS® es una marca registrada de CableLabs® (*Cable Television Laboratories, Inc.*), consorcio de investigación y desarrollo sin ánimos de lucro.

CAPÍTULO I. MARCO TEÓRICO Y CONCEPTUAL.

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.[1]

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).

1.2.5 Arquitectura.

Implementar DOCSIS implica la puesta en funcionamiento de un equipo denominado Sistema de Terminación de Cable Módems (CMTS) en la cabecera, que controla los puertos de envío y recepción de datos hacia la red de cable. Asimismo, en el extremo del cliente debe colocarse un dispositivo que se conoce como Módem de Cable (CM).



Figura 4: Arquitectura de una red HFC con DOCSIS.

Tal como se muestra en la figura 1.4, el CMTS básicamente conecta Internet y la red interna del proveedor con la red HFC.[4]

Para dar soporte a la red DOCSIS, deben emplearse según las especificaciones, dos sistemas fundamentales: el Sistema de Gestión de Red (NMS) y el Sistema de Aprovisionamiento (*Provisioning System*).

Aprovisionar la red significa ofrecer la información necesaria para hacer la red completamente operativa. La red funciona según acuerdos de nivel de servicio entre el abonado y el proveedor, que se manifiestan en un archivo de configuración que el CM de cada abonado obtendrá de un servidor, y que establecerá en la interfaz del abonado las garantías acordadas en el contrato,

CAPÍTULO I. MARCO TEÓRICO Y CONCEPTUAL.

tales como velocidad de datos en bajada, en subida, tasas diferenciadas para audio y video, restricciones, entre otros.

El encargado de aprovisionar la red es el Sistema de Aprovisionamiento, que se conforma de un conjunto de aplicaciones que intervienen en la puesta en operación de los CMs. Este grupo de aplicaciones incluye:

- **Servidor DHCP:** provee 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:** proporciona la referencia de tiempo para la red.

1.2.6 Características y Funcionamiento General.

DOCSIS funciona principalmente entre las capas 1 y 2 del modelo OSI de la ISO. 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.[8]

El espectro total es dividido en una parte para el *upstream* desde 5 MHz hasta 65 MHz, 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.[8]

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 1.1 se muestran las diferentes tasas que pueden alcanzarse.[8]

CAPÍTULO I. MARCO TEÓRICO Y CONCEPTUAL.

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 6 MHz.

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.[1]

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.[8]

En la banda de *upstream*, DOCSIS emplea fundamentalmente el esquema de acceso FDMA/TDMA, conocido como TDMA. 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.[8]

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 que son administradas por el sistema de aprovisionamiento de la red.[8]

1.2.7 Operación básica del protocolo.

La configuración elemental de una red DOCSIS se muestra en la figura 1.5.

CAPÍTULO I. MARCO TEÓRICO Y CONCEPTUAL.

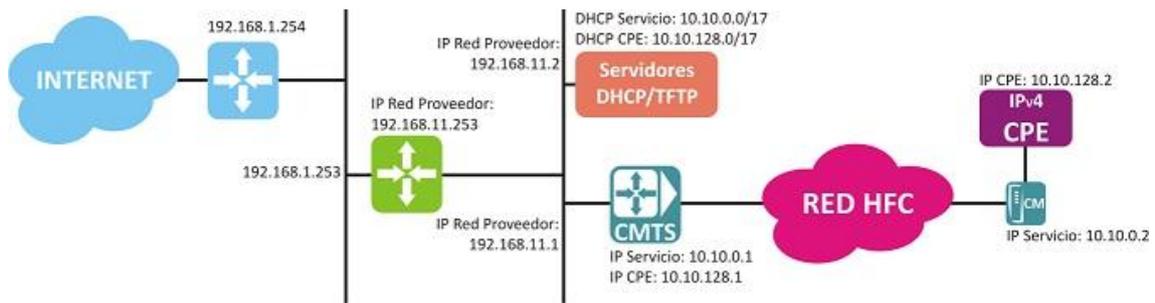


Figura 5: Componentes básicos de una Red DOCSIS.

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 cables módems a nivel de servicio, y otro para los equipos de premisas del cliente (CPE).[9]

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-ecualización. 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.[10]

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

CAPÍTULO I. MARCO TEÓRICO Y CONCEPTUAL.

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.[11]

1.3 Aprovisionamiento en redes DOCSIS.

El proceso de aprovisionamiento en redes DOCSIS se encarga de aportar a los dispositivos la información que estos necesitan para poder realizar sus funciones dentro de una red DOCSIS. En este intervienen dos protocolos fundamentales que son estrictamente necesarios para realizar las actividades de puesta en operación de la red: DHCP y TFTP.

1.3.1 Protocolo de Configuración Dinámica de Host.

DHCP (Dynamic Host Configuration Protocol - Protocolo de configuración dinámica de host) es un protocolo de red que permite a los clientes de una red IP obtener sus parámetros de configuración automáticamente. Se trata de un protocolo de tipo cliente servidor en el que generalmente un servidor posee una lista de direcciones IP dinámicas y las va asignando a los clientes conforme éstas van estando libres, sabiendo en todo momento quién ha estado en posesión de esa IP, cuánto tiempo la ha tenido y a quién se la ha asignado después. Este protocolo se publicó en octubre de 1993, estando documentado actualmente en la RFC 2131. Para DHCPv6 se publica el RFC 3315.[12]

Sin DHCP, cada dirección IP debe configurarse manualmente en cada dispositivo y, si este se mueve a otra subred, se debe configurar otra dirección IP diferente. El DHCP le permite al administrador supervisar y distribuir de forma centralizada las direcciones IP necesarias y, automáticamente, asignar y enviar una nueva IP si fuera el caso en el dispositivo es conectado en un lugar diferente de la red.

En ambientes DOCSIS, DHCP se encarga de proveer a los CMs y los CPEs sus parámetros IP. A los primeros, le entrega la dirección IP que los identificará

CAPÍTULO I. MARCO TEÓRICO Y CONCEPTUAL.

dentro de la red y su máscara, la del servidor de archivos de configuración TFTP, la del servidor de tiempo ToD, la de la frontera del CMTS o puerta de enlace, la del servidor de notificaciones y el nombre del archivo de configuración. A los CPE les entrega su dirección IP, máscara y puerta de enlace, así como los DNS de existir.

1.3.2 Protocolo de Transferencia de Ficheros Trivial.

TFTP (Trivial file transfer Protocol) Protocolo de transferencia de archivos trivial). Es un protocolo de transferencia muy simple semejante a una versión básica de FTP (**F**ile **T**ransfer **P**rotocol), TFTP a menudo se utiliza para transferir pequeños archivos entre ordenadores en una red.

Algunos detalles del TFTP:

- Utiliza UDP (**U**ser **D**atagram **P**rotocol) (en el puerto 69) como protocolo de transporte (a diferencia de FTP que utiliza el puerto 21 TCP **T**ransmission **C**ontrol **P**rotocol).
- No puede listar el contenido de los directorios.
- No existen mecanismos de autenticación o cifrado.
- Se utiliza para leer o escribir archivos de un servidor remoto.

En DOCSIS es el que se encarga de servir los archivos de configuración de los CMs y las actualizaciones de software de los dispositivos, por lo que al igual que el servicio de DHCP son imprescindibles para que estas redes operen.

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

Para la generación de los archivos de configuración el método empleado consiste en la utilización de un editor de texto común para escribir en cierto lenguaje las instrucciones que un generador traduce en un archivo binario, que posteriormente debe ser colocado en la carpeta del servidor TFTP de forma manual, empleando gran cantidad de tiempo.

Por su parte la configuración del servicio DHCP se hace igualmente empleando un editor de texto para modificar el archivo dhcpd.conf. Después de cada modificación, el servicio debe ser reiniciado desde una consola.

CAPÍTULO I. MARCO TEÓRICO Y CONCEPTUAL.

Es de señalar que no existe validación en estos modos de trabajo excepto el ensayo y el error.

1.5 Análisis de los Software existentes.

En lo concerniente a la gestión de equipamiento DOCSIS, aparecen softwares propietarios de algunos fabricantes por un lado, y por otro los que sólo producen software y los de código abierto o libres.

Entre los principales software de Aprovisionamiento, que incluyen la gestión DOCSIS Comerciales 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, y constituyen muchas veces el renglón fundamental de utilidades, por encima del equipamiento.

También existen algunas herramientas que se dedican específicamente al monitoreo y prueba, algunas son:

- Avera.
- Excentis DOCSIS 3.0.
- Fanfare Software iTest.
- iGLASS (cobro mensual).
- Simplesoft DOCSIS OSS Cable Modem Test Suites.
- Tollgrade Communications.
- ZCorum TruVizion.

En cuanto a los proyectos de código abierto u Open Source, se tiene:

- Autoboot DHCP/TFTP provisioning services.

- Birds-Eye.Net Broadband Script Archive.
- Braa Very fast SNMP stack.
- DocsDiag.
- ISG Tools.
- Quelea.

1.6 Metodología de Desarrollo de Software.

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.[13]

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.[13]

1.6.1 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 definido, 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. Además, las metodologías tradicionales no se adaptan adecuadamente a los cambios, por lo que no son métodos adecuados cuando se trabaja en un entorno, donde los requisitos no pueden predecirse o bien pueden variar.[14]

Entre las metodologías tradicionales o pesadas se puede citar:

- RUP (RationalUnifiedProcces).
- MSF (Microsoft Solution Framework).
- Win-Win Spiral Model.
- Iconix.

1.6.2 Metodologías Ágiles.

Los procesos ágiles son una buena elección cuando se trabaja con requisitos desconocidos o variables. Si no existen requisitos estables, no existe una gran posibilidad de tener un diseño estable y de seguir un proceso totalmente planificado, que no vaya a variar ni en tiempo ni en dinero. En estas situaciones, un proceso adaptativo será mucho más efectivo que un proceso predictivo. Por otra parte, los procesos de desarrollo adaptativos también facilitan la generación rápida de prototipos y de versiones previas a la entrega final, lo cual agrada al cliente.[15]

Las metodologías ágiles proporcionan una serie de pautas y principios junto a técnicas pragmáticas que puede que no curen todos los males pero harán la entrega del proyecto menos complicada y más satisfactoria tanto para los clientes como para los equipos de entrega.[15]

Principios del desarrollo ágil de Software

- La prioridad es satisfacer al cliente mediante tempranas y continuas entregas de software que le aporte un valor.
- Dar la bienvenida a los cambios. Se capturan los cambios para que el cliente tenga una ventaja competitiva.
- Entregar frecuentemente software que funcione desde un par de semanas a un par de meses, con el menor intervalo de tiempo posible entre entregas.
- La gente del negocio y los desarrolladores deben trabajar juntos a lo largo del proyecto.
- Construir el proyecto en torno a individuos motivados. Darles el entorno y el apoyo que necesitan y confiar en ellos para conseguir finalizar el trabajo.
- El diálogo cara a cara es el método más eficiente y efectivo para comunicar información dentro de un equipo de desarrollo.
- .El software que funciona es la medida principal de progreso
- .Los procesos ágiles promueven un desarrollo sostenible. Los promotores, desarrolladores y usuarios deberían ser capaces de mantener una paz constante.

CAPÍTULO I. MARCO TEÓRICO Y CONCEPTUAL.

- La atención continua a la calidad técnica y al buen diseño mejora la agilidad.
- .La simplicidad es esencial.
- Las mejores arquitecturas, requisitos y diseños surgen de los equipos organizados por sí mismos.
- .En intervalos regulares, el equipo reflexiona respecto a cómo llegar a ser más efectivo, y según esto ajusta su comportamiento.[15]

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).

Después del análisis de las distintas 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.
- 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 está contenida en 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

CAPÍTULO I. MARCO TEÓRICO Y CONCEPTUAL.

quiere ver completados y los hace del conocimiento del equipo. Luego, 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.
- Un principio clave de Scrum es el reconocimiento que durante un proyecto los clientes pueden cambiar de idea sobre lo que quieren y necesitan (a menudo llamado requirement churn), 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.
- 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.[13]

1.7 Lenguajes utilizados.

1.7.1 Lenguaje de Modelado.

El Lenguaje de Modelado Unificado Unified Modeling Language (**UML**) 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.[16]

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

CAPÍTULO I. MARCO TEÓRICO Y CONCEPTUAL.

lenguaje de modelado visual que permite una abstracción del sistema y sus componentes.[16]

Entre sus objetivos fundamentales se encuentran:

- 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.[16]

1.7.2 Lenguaje de Programación.

Los lenguajes de programación son idiomas artificiales diseñados para expresar cálculos y procesos que serán llevados a cabo por ordenadores. Un lenguaje de programación está formado por un conjunto de palabras reservadas, símbolos y reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones. El proceso de programación consiste en la escritura, compilación y verificación del código fuente de un programa.[17]

Entre los lenguajes de programación que existen actualmente se pueden mencionar los siguientes:

- Java
- C++
- C
- Python
- Perl

Luego de mencionar algunos de los diferentes lenguajes de programación se elige **Java** como lenguaje de programación por las siguientes características:

CAPÍTULO I. MARCO TEÓRICO Y CONCEPTUAL.

- **Orientado a objetos;** La primera característica, orientado a objetos (“OO”), se refiere a un método de programación y al diseño del lenguaje. Aunque hay muchas interpretaciones para OO, una primera idea es diseñar el software de forma que los distintos tipos de datos que usen estén unidos a sus operaciones. Así, los datos y el código (funciones o métodos) se combinan en entidades llamadas objetos. Un objeto puede verse como un paquete que contiene el “comportamiento” (el código) y el “estado” (datos). El principio es separar aquello que cambia de las cosas que permanecen inalterables. Frecuentemente, cambiar una estructura de datos implica un cambio en el código que opera sobre los mismos, o viceversa. Esta separación en objetos coherentes e independientes ofrece una base más estable para el diseño de un sistema software. El objetivo es hacer que grandes proyectos sean fáciles de gestionar y manejar, mejorando como consecuencia su calidad y reduciendo el número de proyectos fallidos. Otra de las grandes promesas de la programación orientada a objetos es la creación de entidades más genéricas (objetos) que permitan la reutilización del software entre proyectos, una de las premisas fundamentales de la Ingeniería del Software. Un objeto genérico “cliente”, por ejemplo, debería en teoría tener el mismo conjunto de comportamiento en diferentes proyectos, sobre todo cuando estos coinciden en cierta medida, algo que suele suceder en las grandes organizaciones. En este sentido, los objetos podrían verse como piezas reutilizables que pueden emplearse en múltiples proyectos distintos, posibilitando así a la industria del software a construir proyectos de envergadura empleando componentes ya existentes y de comprobada calidad; conduciendo esto finalmente a una reducción drástica del tiempo de desarrollo.[18]
- **Simple;** Java reduce hasta en un 50% los errores más comunes de programación respecto a otros lenguajes (elimina la Aritmética de punteros, Registros, Macros, definición de tipos, liberar memoria entre otros).[18]

CAPÍTULO I. MARCO TEÓRICO Y CONCEPTUAL.

- **Distribuido;** Java proporciona librerías y herramientas para que los programas sean distribuidos, es decir, para que se ejecuten en varias máquinas e interactuando entre ellas.[18]
- **Multiplataforma Plataforma (arquitectura neutral);** las aplicaciones java (.java) para poder ser ejecutadas, antes deberán ser compiladas dando como resultado un código intermedio denominado bytecode (.class). Los bytecode son ejecutados por la Java Virtual Machine (JVM) donde son interpretados y luego traducidos al lenguaje máquina según la plataforma donde se ejecute. La ventaja de la JVM es la portabilidad del lenguaje. Puede ser interpretado y ejecutado en cualquier, ya sea Linux o Mac, solo dependerá de que dichos entornos posean las JVM específicos para cada uno. Multiplataforma, significa que programas escritos en el lenguaje Java pueden ejecutarse igualmente en cualquier tipo de hardware. Este es el significado de ser capaz de escribir un programa una vez y que pueda ejecutarse en cualquier dispositivo, tal como reza el axioma de Java, —*write once, run anywhere*—. Para establecer Java como parte principal de la red, el compilador Java compila su código a un fichero objeto de formato independiente de la arquitectura del ordenador donde se ejecutará. Cualquier ordenador que posea el sistema de ejecución Java Run Time (JRE) podrá ejecutar ese código objeto. El código fuente Java se “compila” a código bytes (bytecode) de alto nivel independiente de la máquina donde se ejecute.[18]
- **Es Robusto: recolección de basura;** Java realiza verificaciones en busca de problemas tanto en tiempo de compilación como en tiempo de ejecución, estas comprobaciones ayudan a detectar errores en ciclo de desarrollo: obligando la declaración explícita de métodos. Además, se encarga de la liberación de memoria. En Java el problema de las fugas de memoria se evita en gran medida gracias a la recolección de basura (*automatic garbage collector*). El programador determina cuándo se crean los objetos y el entorno en tiempo de ejecución de Java (*Java Runtime*) es el responsable de gestionar el ciclo de vida de los objetos. El programa, u otros objetos pueden tener localizado un objeto mediante

CAPÍTULO I. MARCO TEÓRICO Y CONCEPTUAL.

una referencia a éste. Cuando no quedan referencias a un objeto, el recolector de basura de Java borra el objeto, liberando así la memoria que ocupaba previniendo posibles fugas (ejemplo: un objeto creado y únicamente usado dentro de un método sólo tiene entidad dentro de éste; al salir del método el objeto es eliminado). Aun así, es posible que se produzcan fugas de memoria si el código almacena referencias a objetos que ya no son necesarios, es decir, pueden aún ocurrir, pero en un nivel conceptual superior. En definitiva, el recolector de basura de Java permite una fácil creación y eliminación de objetos, mayor seguridad y puede que más rápida que en C++.[18]

- ➔ **Es Seguro;** a nivel de lenguaje, se eliminan punteros y casting implícito de los compiladores de lenguajes convencionales para prevenir el acceso ilegal a la memoria. A nivel de ejecución, el código Java pasa una serie de pruebas (test) antes de ejecutarse en la máquina, pasando el código a través de un verificador de código bytecode que comprueba el formato de los fragmentos de código y aplica un verificador de teoremas para detectar fragmentos de código ilegal (código que falsea punteros, viola derechos de acceso sobre objetos o intenta cambiar el tipo de clase de un objeto). Las aplicaciones Java son seguras porque no acceden a zonas delicadas de memoria o de sistema, por lo que evitan la interacción de ciertos tipos de virus. A nivel de código fuente, el JDK proporciona un desensamblador de bytecode que permite que cualquier aplicación (programa desarrollado en java) pueda ser convertido a código fuente, esta es sin duda una desventaja para el desarrollador, no obstante, existen alternativas para impedir que el código sea expuesto.[18]
- ➔ **Es portable;** más allá de la portabilidad intrínseca a la multiplataforma (arquitectura neutral), Java implementa otros estándares de portabilidad, p.e. los enteros son siempre enteros (32bits) y Java construye sus interfaces de usuario a través de un sistema abstracto de ventanas de manera que las ventanas pueden ser implementadas en entornos Unix, Mac o Windows.[18]

CAPÍTULO I. MARCO TEÓRICO Y CONCEPTUAL.

- **Es interpretado;** El intérprete de java (sistema run time) puede ejecutar directamente el código objeto, sin embargo, debido a que no existen JDK específicos para las diversas plataformas, java es lento respecto a otros lenguajes ya que primero deberá ser interpretado y no ejecutado como en C++. El proceso consiste: i) primero se compila el código fuente escrito en lenguaje Java, para generar un código conocido como “bytecode” (específicamente Java bytecode) instrucciones máquina simplificadas específicas de la plataforma Java; ii) Esta pieza está “a medio camino” entre el código fuente y el código máquina que entiende el dispositivo destino; iii) El bytecode es ejecutado entonces en la máquina virtual (JVM), un programa escrito en código nativo de la plataforma destino (que es el que entiende su hardware), que interpreta y ejecuta el código. Además, se suministran bibliotecas adicionales para acceder a las características de cada dispositivo (como los gráficos, ejecución mediante hilos o threads, la interfaz de red) de forma unificada. Se debe tener presente que, aunque hay una etapa explícita de compilación, el bytecode generado es interpretado o convertido a instrucciones máquina del código nativo por el compilador JIT (Just In Time).[18]
- **Es Multihilo (multithreaded);** Java permite muchas funciones simultáneas en una aplicación, al respecto, los hilos son procesos o piezas independientes dentro de un gran proceso; al ser éstos hilos construidos en el mismo lenguaje, son más fáciles de usar y más robustos respecto a otros lenguajes. Al ser multihilo, se tiene mejor rendimiento interactivo y mejor comportamiento en tiempo real (pero limitado a las capacidades del Sistema Operativo, OS subyacente), p.e. al navegar, java puede acceder a la información de la página sin tener que esperar a que el navegador cargue tediosamente figuras u animaciones.[18]
- **Es Dinámico;** Java no intenta conectar todos los módulos que comprenden una aplicación hasta el mismo tiempo de ejecución. Las nuevas librerías no paralizan el tiempo de ejecución siempre que contengan el API anterior. También simplifica el uso de protocolos

CAPÍTULO I. MARCO TEÓRICO Y CONCEPTUAL.

nuevos o actualizados; si uno ejecuta una aplicación java sobre la red, java es capaz de “importar” automáticamente cualquier pieza que el sistema necesite para funcionar y, para evitar “importar” cada vez, java implementa las opciones de persistencia para que no se eliminen cuando se limpie el caché de la máquina.[18]

- De momento, es público. Puede conseguirse un JDK (Java Developer's Kit) o Kit de desarrollo de aplicaciones Java gratis.[18]

1.7.3 Lenguaje 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:[19]

- 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.[20]

1.8 Sistemas Gestores de Base de Datos.

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

CAPÍTULO I. MARCO TEÓRICO Y CONCEPTUAL.

flotante, enteros, cadenas de caracteres, cantidades monetarias y fechas. Está generalmente reconocido que este modelo será inadecuado para las aplicaciones futuras de procesamiento 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. [11]

Existen distintos gestores de base de datos como son:

- PostgreSQL.
- MySQL.

Después de mencionar distintos gestores de base de datos se elige **Postgres** porque 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:

- Clases.
- Herencia.
- Tipos.
- Funciones.[21]

También posee características que aportan potencia y flexibilidad adicional:

- Restricciones (Constraints).
- Disparadores (triggers).
- Reglas (rules).
- Integridad transaccional.
- Soporta casi toda la sintaxis SQL (incluyendo subconsultas, transacciones, tipos y funciones definidas por el usuario).
- Cuenta con un amplio conjunto de enlaces con lenguajes de programación (C, C++, java, Perl, tcl, python, etc.).[21]

1.9 Herramientas Utilizadas.

Los entornos de desarrollo integrado (IDE) que permiten editar programas en java, compilarlos, ejecutarlos, depurarlos y construir rápidamente el interfaz

CAPÍTULO I. MARCO TEÓRICO Y CONCEPTUAL.

gráfico de una aplicación eligiendo los componentes, facilitan la escritura de aplicaciones java, proporcionando una serie de servicios comunes, que a su vez están disponibles a través del IDE.[22]

Entre los diferentes IDE se encuentran:

- NetBeans.
- Eclipse.

De los IDE mencionados se elige NetBeans por las siguientes características:

- Es una herramienta para programadores pensada para escribir, compilar, depurar y ejecutar programas. Está escrito en Java, pero puede servir para cualquier otro lenguaje de programación. Existe además un número importante de módulos para extender el IDE NETBEANS. El IDE NETBEANS es un producto libre y gratuito sin restricciones de uso.[22]

Es de código abierto escrito completamente en Java usando la plataforma NETBEANS. Soporta el desarrollo de todos los tipos de aplicación Java (J2SE, web, EJB y aplicaciones móviles). Entre sus características se encuentra un sistema de proyectos basado en Ant, control de versiones y refactoring.[22]

1.8.1 Visual Paradigm.

Es una herramienta CASE (Computer Aided Software Engineering – Ingeniería de Software Asistida por Computadora). 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.[23]

Ha sido concebida para soportar el ciclo de vida completo del proceso de desarrollo del software a través de la representación de todo tipo de diagramas. Constituye una herramienta privada disponible en varias ediciones, cada una destinada a unas necesidades: Enterprise, Professional, Community, Standard, Modeler y Personal. Existe una alternativa libre y gratuita de este software, la versión Visual Paradigm UML 6.4 Community Edition. Fue diseñado para una amplia gama de usuarios interesados en la construcción de sistemas de

CAPÍTULO I. MARCO TEÓRICO Y CONCEPTUAL.

software de forma fiable a través de la utilización de un enfoque orientado a objetos.[23]

1.9 Conclusiones del Capítulo.

En este capítulo se realizó un estudio de los diferentes conceptos relacionados al dominio del problema, así como un análisis de las diferentes tecnologías y herramientas a utilizar en el software propuesto.

Se seleccionó la metodología Scrum, como guía para la documentación del software propuesto y UML como lenguaje para modelar el análisis y diseño.

Se seleccionó además como lenguaje de programación Java, usando los servidores DHCP, TFTP, ToD.

Se seleccionó PostgreSQL como gestor de base de datos.

Se eligió NetBeans, Visual Paradigm, como herramientas de desarrollo.

Esta elección fue realizada en base a las potencialidades de cada una de las herramientas y lenguajes para realizar la implementación del sistema informático propuesto.

El sistema informático propuesto propone la implementación de un Sistema de Recomendación basado en conocimiento en el que se tendrán en cuenta las necesidades de los usuarios y el grado de importancia de los softwares que necesite.

CAPÍTULO 2. DESCRIPCIÓN Y CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA.

2.1 Introducción al Capítulo.

En este capítulo se utiliza como metodología de desarrollo SCRUM en la cual se define la pila del producto donde van a estar incluidos los requisitos funcionales del sistema, la pila de los sprints 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. Además el diagrama de casos de uso del sistema y el modelo lógico y físico de la base de datos.

2.2 Pila del Producto.

La pila de producto es el corazón de Scrum. Es donde empieza todo. La Pila de Producto es, básicamente, una lista priorizada de requisitos, historias, o funcionalidades. Cosas que el cliente quiere, descritas usando la terminología del cliente. Llamamos a esto historias, o a veces simplemente elementos de la Pila.

Es el inventario de características que el propietario del producto desea obtener, ordenado por orden de prioridad.

En la pila del producto se incluyen los campos:

ID: identificador único, simplemente un número auto-incremental.

Importancia: importancia que el Dueño de Producto da a esta historia. Por ejemplo, 10 ó 150. Mientras más alto, más importante es la historia.

Estimación inicial: valoración inicial del Equipo acerca de cuanto trabajo es necesario para implementar la historia, comparada con otras historias.

Nombre: descripción corta de la historia. Suficientemente clara como para que el Dueño de Producto comprenda aproximadamente de qué se está hablando, y suficientemente clara como para distinguirla de las otras historias. Normalmente, 2 a 10 palabras.

Criterio de validación: descripción a alto nivel de cómo se demostrará esta historia al final del Sprint.

CAPÍTULO 2. DESCRIPCIÓN Y CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA.

Observaciones: es cualquier otra información, clarificación, referencia a otras fuentes de información, etc. Normalmente muy breve.[25]

Pila del Producto:

Id	Nombre	Importancia	Estimación inicial	Como probarlo	Observaciones
1	Autenticar Usuario	30	3	El usuario selecciona en el menú de autenticación en donde tiene que poner usuario y contraseña, en caso de ser correctas podrá acceder a las opciones según su privilegio, en caso de ser incorrectas un mensaje de error y no puede acceder a la información.	Requiere de un algoritmo de encriptación para las contraseñas. Se necesita hacer consultas a la base de datos
2	Gestionar Usuario	35	8	El personal encargado selecciona en el menú de gestionar usuario y escoge la opción que desee realizar como insertar, modificar o eliminar usuario, en caso de crear debe asignarle los privilegios del mismo.	Se necesitan hacer consultas a la base de datos
3	Cambiar Contraseña	30	4	Permite cambiar la contraseña, se llenan los campos del usuario contraseña vieja y contraseña nueva, en caso de que el usuario y la contraseña vieja sean correctas entonces la contraseña cambia sino un	Se necesitan hacer consultas a la base de datos.

CAPÍTULO 2. DESCRIPCIÓN Y CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA.

				mensaje de error.	
4	Gestionar Fichero DHCP	40	15	Asignar a cada equipo DOCSIS que se conecte a la red y este previamente en el inventario, una dirección de IP que le entregará el servidor de DHCP y un archivo de configuración.	Cada vez que se haga cualquier cambio en el fichero DHCP hay que reiniciar el mismo para que los cambios sean eficientes. Se necesitan hacer consultas a la base de datos
5	Gestionar Inventario de equipos DOCSIS	30	6	El personal encargado accede al menú de gestionar inventario de equipos DOCSIS donde puede Insertar, Modificar y Eliminar el inventario de equipos DOCSIS.	Se necesitan hacer consultas a la base de datos
6	Gestionar Archivo de Configuración	45	15	El personal encargado accede al menú de gestionar archivo de configuración donde puede Insertar, Modificar y eliminar los archivos de configuración.	Requiere de un algoritmo de encriptación para comprobar que no fue dañado. Se necesita hacer

CAPÍTULO 2. DESCRIPCIÓN Y CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA.

					consultas a la base de datos
7	Visualizar localización de los equipos	35	11	Muestra gráficamente la localización de todos los equipos existentes en un mapa de referencia.	Se necesitan hacer consultas a la base de datos
8	Cerrar sesión	40	4	El usuario selecciona la opción Salir y el sistema muestra la pantalla de inicio de autenticación de usuario.	
9	Gestionar ubicación	40	5	El personal encargado accede al menú de gestionar ubicación donde puede Insertar, Modificar y eliminar las ubicaciones.	

Tabla 2: Pila del Producto

2.3 Requerimientos del Sistema.

Son una condición y exigencia que debe cumplir el sistema que se va a desarrollar a solicitud del usuario o cliente. Existen dos tipos de requisitos del sistema:

- Requisitos Funcionales: Es un área de funcionalidad que debe de soportar el sistema.
- Requisitos No Funcionales: Son propiedades o cualidades que el producto debe tener.

2.3.1 Requisitos No Funcionales.

Los requerimientos no funcionales del sistema propuesto son los siguientes:

Requisitos de Software

- Sistema Operativo Linux con ambiente gráfico.
- JRE 6 o superior.

CAPÍTULO 2. DESCRIPCIÓN Y CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA.

- Servicio PostgreSQL. 9.0
- Servicio DHCP3-Server
- Servicio TFTP3-Server

Requisitos de Hardware

- Procesador dual Core a 2.4 GHz o superior.
- RAM 2GB o superior.
- 10 GB de espacio en Disco Duro.
- Tarjeta de Red Ethernet.
- Dispositivos de Interfaces Humanas: Mouse Teclado.
- Monitor con resolución 1600x900 mínima.

Requisitos de Seguridad

Este es quizás el tipo de requerimiento más difícil, que provocará los mayores riesgos si no se maneja correctamente.

- **Confidencialidad:** La información manejada por el sistema está protegida de acceso no autorizado con la asignación de privilegios a los usuarios para que solamente el personal autorizado pueda acceder a la información de acceso no autorizado.
- **Disponibilidad:** A los usuarios autorizados se les garantiza el acceso a la información y que los mecanismo de utilizados para lograr la seguridad no ocultarán o retrasarán a los usuarios para obtener los datos deseados en un momento dado.
- **Integridad:** La información manejada por el sistema será objeto de cuidadosa protección contra la corrupción y estados inconsistentes mediante la utilización de algoritmos de encriptación.

La seguridad es un requerimiento no funcional que genera posiblemente requisitos funcionales. ¿Cuáles requisitos? Depende de la propuesta de seguridad que se tenga para el sistema.

Requisitos de Usabilidad

CAPÍTULO 2. DESCRIPCIÓN Y CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA.

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 descansan en las bases de diseño centrado en el usuario.

Facilidad de comprensión:

- Las funcionalidades del sistema deben ser fáciles de ubicar.
- Permitir al usuario pedir ayuda en cualquier momento.
- El tamaño y color de la letra debe permitir leer con facilidad.
- La cantidad de colores en la pantalla debe ser adecuada para el tipo de información que contiene.
- Permitir al usuario parar el programa y salir de él en cualquier momento.

Capacidad de Uso.

- La ayuda para el uso del software se deben hallar rápido y fácilmente.

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.

Requisitos de Fiabilidad.

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

Recuperación.

- El software debe recuperarse fácilmente después de una caída o falla.
- Se debe permitir a los usuarios trabajar con el producto de software el tiempo necesario.

CAPÍTULO 2. DESCRIPCIÓN Y CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA.

Tolerancia a fallas

- El software debe presentar una explicación y opciones para la solución, cuando ocurre la falla.
- Las opciones de solución a fallas deben resolver el problema.

Requisitos 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 e integridad.

Requisitos 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 Cayo Santa María, Santa Clara donde prestan sus servicios. 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.

Requisitos Políticos-Culturales.

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

Requisitos Legales

La herramienta propuesta responde a los intereses de la Empresa Telecable Internacional.

CAPÍTULO 2. DESCRIPCIÓN Y CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA.

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. 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 permitirsele.[25] Para que una planificación de Sprint se produzca correctamente debe cumplir los siguientes requisitos:

- Una meta de Sprint
- Una lista de miembros (y su nivel de dedicación, si no es del 100%)
- Una Pila de Sprint (lista de historias incluidas en el Sprint)
- Una fecha concreta para la Demo del Sprint
- Un lugar y momento definidos para el Scrum Diario.[25]

2.4.1 Listado de los Sprint.

Número de Sprint	Duración(días)	Participantes	Factor de Dedicación
Sprint 1	15	Devis González Alejandro León Yaima Toledo	0.70
Sprint 2	15	Devis González Alejandro León Yaima Toledo	0.71
Sprint 3	15	Devis González Alejandro León Yaima Toledo	0.714
Sprint 4	15	Devis González Alejandro León Yaima Toledo	0.68
Sprint 5	15	Devis González	0.65

CAPÍTULO 2. DESCRIPCIÓN Y CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA.

		Alejandro León Yaima Toledo	
--	--	--------------------------------	--

Tabla 3: Listado de los Sprint.

2.4.2 Técnicas de estimación de Sprint.

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

➤ **A 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.

➤ **Cálculo de velocidad basado en días-hombre disponibles y factor de dedicación.**

Esta técnica consta de dos pasos:

- Decidir la velocidad estimada.
- Calcular cuántas historias se pueden añadir sin sobrepasar la velocidad estimada.

La velocidad es una medida de “cantidad de trabajo realizado”, donde cada elemento se evalúa en función de su estimación inicial.[25]

De las dos técnicas señaladas se ha elegido la segunda porque es más eficiente en el momento de calcular la velocidad estimada para un Sprint:

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.

Velocidad estimada de este Sprint:

$(\text{Días-Hombre Disponibles}) \times (\text{Factor de Dedicación}) = \text{Velocidad Estimada}$

El factor de dedicación: es una estimación de cómo de centrado va a estar el equipo. Un factor de dedicación bajo puede significar que el equipo espera

CAPÍTULO 2. DESCRIPCIÓN Y CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA.

encontrar muchas distracciones e impedimentos o que considera que sus propias estimaciones son optimistas.

La mejor manera de determinar un factor de dedicación razonable es estudiar el último Sprint (o incluso mejor, la media de los últimos Sprint).

Factor de Dedicación del Ultimo Sprint.

$$\text{(FACTOR DE DEDICACIÓN)} = \frac{\text{(VELOCIDAD REAL)}}{\text{(DIAS-HOMBRE DISPONIBLES)}}$$

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

2.4.3 Descripción de los Sprint.

Sprint 1:

➤ Metas del Sprint:

- Permitir a cada usuario interactuar con las facilidades que brinda el sistema de acuerdo a los privilegios asignados.
- Insertar, Modificar y eliminar Usuarios.
- Asignarle los privilegios a los usuarios.
- Permitir a los usuarios poder cambiar la contraseña.

➤ Pila del Sprint 1:

Autenticar Usuario
Gestionar Usuario
Cambiar Contraseña

Tabla 4: Pila del Sprint 1.

➤ Estimación de Historias del Sprint.

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

CAPÍTULO 2. DESCRIPCIÓN Y CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA.

Trabajadores	Días-Hombres(disponibles)	Factor de dedicación
Deivis González	12	0.70
Alejandro León	5	0.70
Yaima Toledo	4	0.70

Tabla 5: Estimación de Historia del Sprint 1.

Velocidad Estimada= 21 * 0.70

Velocidad Estimada= 15 (puntos de historia).

➤ Historias Incluidas en el Sprint

- Autenticar Usuario – 3 puntos de historia inicialmente, utilizando la técnica de ojo de buen cubero.
- Gestionar Usuario- 8 puntos de historia inicialmente, utilizando la técnica de ojo de buen cubero.
- Cambiar contraseña -4 puntos de historia inicialmente, utilizando la técnica de ojo de buen cubero.

➤ Como Probar cada Historia del Sprint.

➤ Autenticar Usuario.

Se accede al menú de autenticación donde se muestra la interfaz de autenticarse en la que el usuario debe poner su usuario y contraseña, en caso de ser correctos podrá acceder a las opciones según sus privilegios de administración. En caso contrario se notificara un mensaje de error.

➤ Gestionar Usuario.

Se accede al menú Gestionar Usuario, donde escoge la opción que desee realizar, tanto eliminar, modificar, como crear un usuario. Cuando crea un usuario tiene que asignarle los privilegios, si modifica puede cambiar el usuario, la contraseña, y los privilegios y para eliminar un usuario se muestra la lista de los usuarios existentes, selecciona los que desea eliminar y da clic en el botón de eliminar y los usuarios serán eliminados, en cualquiera de las tres opciones en caso de no haber conexión con la base de datos se mostrara un mensaje con dicha información.

➤ Cambiar Contraseña.

CAPÍTULO 2. DESCRIPCIÓN Y CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA.

Se accede al menú Cambiar Contraseña, donde se muestra la interfaz de cambiar la contraseña, el usuario llena los campos que se piden para realizar la acción deseada y la contraseña se habrá cambiado correctamente, en caso de errores como que el usuario actual y la contraseña actual no coincidan se mostrara un mensaje de error.

➤ Lista de Miembros.

Deivis González – 60% de trabajo en el Sprint.

Alejandro León – 20% de trabajo en los Sprint.

Yaima Toledo – 20% de trabajo en los Sprint.

➤ Un lugar y momento definido para el Scrum Diario.

Laboratorio C de Informática – 9.00 a.m.

➤ Historias divididas en tareas.

Iniciar sesión.	Crear interfaz gráfica.
	Autenticar usuario.
Gestionar Usuario.	Crear interfaz gráfica.
	Insertar, modificar y eliminar usuario.
	Validar mensajes de salida
Cambiar Contraseña	Crear interfaz gráfica.
	Insertar dato para cambiar contraseña.

Tabla 6: Historias divididas en tareas Sprint 1.

Sprint 2.

➤ Metas del Sprint.

- Cuando se modificar el fichero DHCP lo que se hace es asignar una MAC, una dirección de IP y el fichero de configuración de cada equipo DOCSIS que se conecte a la red y esté previamente en el inventario.

➤ Pila del Sprint 2.

CAPÍTULO 2. DESCRIPCIÓN Y CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA.

Gestionar Fichero DHCP

Tabla 7: Pila del Sprint 2.

➤ 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-hombre disponibles y factor de dedicación.

Factor de Dedicación = 15 / 21 = 0.71

Trabajadores	Días-Hombres(disponibles)	Factor de dedicación
Deivis González	13	0.71
Alejandro León	4	0.71
Yaima Toledo	4	0.71

Tabla 8: Cálculo de la velocidad estimada para el Sprint 2

Velocidad Estimada = 21 * 0.71

Velocidad Estimada = 15 (puntos de historia)

➤ Historias Incluidas en el Sprint.

- Modificar Fichero DHCP – 15 puntos de historia inicialmente utilizando la técnica de ojo buen cubero.

➤ Como probar cada Historia del Sprint.

- Modificar Fichero DHCP.

El personal encargado de administrar dicha información después de haber introducido usuario y contraseña selecciona en el menú la opción de Modificar Fichero DHCP, en la interfaz que se le muestra inserta los datos relacionados y da clic en el botón de modificar, en caso de ser eficiente muestra un mensaje fichero modificado y en caso contrario un mensaje de error.

➤ Lista de Miembros.

- Deivis González – 60% de trabajo en el Sprint.
- Alejandro León – 20% de trabajo en los Sprint.

CAPÍTULO 2. DESCRIPCIÓN Y CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA.

➤ Yaima Toledo – 20% de trabajo en los Sprint.

➤ Un lugar y momento definidos para el Scrum Diario.

Laboratorio C de Informática – 9.00 a.m.

➤ Historias divididas en tareas.

Modificar fichero de DHCP.	Crear interfaz gráfica.
	Modificar fichero de DHCP.
	Validar mensaje de salida

Tabla 9: Historias divididas en tareas Sprint 2

Sprint 3

➤ Metas del Sprint.

- Insertar, modificar y eliminar inventario de equipos DOCSIS.
- Brindar reportes de asignación y ubicación de direcciones de IP.
- Habilitar/Deshabilitar y observar el estado del servidor TFTP.

➤ Pila del Sprint 3.

Gestionar Inventario de Equipos DOCSIS
Cerrar Sesión
Gestionar Ubicación

Tabla 10: Pila del Sprint 3.

➤ Estimación de Historia del Sprint 3.

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

Factor de dedicación = 15 / 21 = 0.714

Trabajadores	Días-Hombres(disponibles)	Factor de dedicación
Deivis González	13	0.714
Alejandro León	4	0.714
Yaima Toledo	5	0.714

Tabla 11: Cálculo de la velocidad estimada para el Sprint 3

CAPÍTULO 2. DESCRIPCIÓN Y CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA.

Velocidad Estimada = $22 * 0.714$

Velocidad Estimada = 15(puntos de historia)

➤ **Historias Incluidas en el Sprint.**

- Gestionar Inventario de equipos DOCSIS Actividades – 6 puntos de historia inicialmente utilizando la técnica de ojo de buen cubero.
- Cerrar Sesión Actividades – 4 puntos de historia inicialmente utilizando la técnica de ojo de buen cubero.
- Gestionar Ubicación Actividades – 5 puntos de historia inicialmente utilizando la técnica de ojo de buen cubero.

➤ **Como probar cada historia del Sprint.**

- Gestionar Inventario de equipos DOCSIS.

Después que el usuario este autenticado y que tenga privilegios para acceder a este menú entonces puede insertar los equipos DOCSIS llenando los campos que se solicita en caso de ya estar, muestra un mensaje que el equipo ya existe, si está correcta la información muestra un mensaje acción completada y para eliminar se muestran todos los equipos DOCSIS donde el usuario selecciona los que desee eliminar.

- Cerrar Sesión

Entra al menú de cerrar sesión y da clic en terminar sesión y muestra la interfaz de autenticación.

- Gestionar Ubicación.

Después que el usuario esté autenticado y que tenga privilegios para acceder a este menú entonces puede insertar las ubicaciones llenando los campos que requiere en caso de ya estar, muestra un mensaje que la ubicación ya existe, si está correcta la información muestra un mensaje acción completada, para eliminar se muestran todos los equipos DOCSIS donde el usuario selecciona los que desee eliminar.

➤ **Lista de Miembros.**

- Deivis González – 60% de trabajo en el Sprint.
- Alejandro León – 20% de trabajo en los Sprint.

CAPÍTULO 2. DESCRIPCIÓN Y CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA.

➤ Yaima Toledo – 20% de trabajo en los Sprint.

➤ **Un lugar y momento definidos para el Scrum Diario.**

Laboratorio C de Informática – 9.00 a.m.

➤ **Historias divididas en tareas.**

Gestionar equipos DOCSIS.	Crear interfaz gráfica.
	Insertar, modificar y eliminar equipos DOCSIS.
	Validar mensajes de salidas.
Cerrar Sesión	Crear interfaz gráfica.
	Cerrar la sesión y mostrar la interfaz de iniciar sesión
Gestionar Ubicación	Crear interfaz gráfica.
	Insertar, modificar y eliminar Ubicaciones.

Tabla 12: Historias divididas en tareas Sprint 3

.

Sprint 4

➤ **Metas del Sprint.**

➤ Crear, modificar y eliminar el archivo de configuración por el personal autorizado.

➤ **Pila del Sprint 4.**

Gestionar Archivo de Configuración

Tabla 13: Pila del Sprint 4.

➤ **Estimación de Historia del Sprint 4.**

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

Factor de dedicación = 15 / 22 = 0.68

CAPÍTULO 2. DESCRIPCIÓN Y CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA.

Trabajadores	Días-Hombres(disponibles)	Factor de dedicación
Deivis González	13	0.68
Alejandro León	5	0.68
Yaima Toledo	5	0.68

Tabla 14: Cálculo de la velocidad estimada para el Sprint 4

Velocidad Estimada = 23 * 0.68

Velocidad Estimada = 15(puntos de historia)

➤ **Historias Incluidas en el Sprint**

- Gestionar Archivo de Configuración – 15 puntos de historia inicialmente utilizando la técnica de ojo de buen cubero.

➤ **Como probar cada historia del Sprint.**

- Gestionar Archivo de Configuración.

Después que el usuario esté autenticado y tenga privilegios para acceder a este menú entonces podrá crear archivos de configuración entrando los datos necesarios en caso de algún error, se muestra un mensaje de error, podrá modificar los ficheros de configuración donde se le muestra la lista de los ficheros de configuración y selecciona el que va a modificar y eliminar ficheros de configuración seleccionando los que desea eliminar en la lista que se muestra.

➤ **Lista de Miembros.**

- Deivis González – 60% de trabajo en el Sprint.
- Alejandro León – 20% de trabajo en los Sprint.
- Yaima Toledo – 20% de trabajo en los Sprint.

➤ **Un lugar y momento definidos para el Scrum Diario.**

Laboratorio C de Informática – 9.00 a.m.

➤ **Historias divididas en tareas.**

Gestionar archivo de configuración.	Crear interfaz gráfica.
	Insertar, modificar y eliminar archivo

CAPÍTULO 2. DESCRIPCIÓN Y CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA.

	de configuración.
	Validar mensajes de salidas.

Tabla 15: Historias divididas en tareas Sprint 4

Sprint 5

➤ Metas del Sprint.

- Habilitar/Deshabilitar y observar los servicios de ToD
- Mostrar gráficamente la localización de los equipos en un mapa de referencia.

➤ Pila del Sprint 5

Gestionar Servicios de ToD
Visualizar la Localización de los equipos

Tabla 16: Pila del Sprint 5.

➤ Estimación de Historia del Sprint 5.

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

Factor de dedicación = 15 / 23 = 0.65

Trabajadores	Días-Hombres(disponibles)	Factor de dedicación
Deivis González	13	0.65
Alejandro León	6	0.65
Yaima Toledo	5	0.65

Tabla 17: Cálculo de la velocidad estimada para el Sprint 4

Velocidad Estimada = 24 * 0.65

Velocidad Estimada = 15(puntos de historia)

➤ Historias Incluidas en el Sprint

- Gestionar Servicios de ToD– 4 puntos de historia inicialmente utilizando la técnica de ojo de buen cubero.
- Visualizar la localización de los Equipos – 11 puntos de historia inicialmente utilizando la técnica de ojo de buen cubero.

CAPÍTULO 2. DESCRIPCIÓN Y CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA.

➤ Como probar cada historia del Sprint.

➤ Gestionar Servicios de ToD.

Después que el usuario esté autenticado y tenga los privilegios para acceder a este menú entonces podrá Habilitar/Deshabilitar y observar los servicios de ToD.

➤ Visualizar la localización de los Equipos.

Muestra gráficamente la localización de los equipos en un mapa de referencia para saber dónde están ubicados o la cantidad que hay en disposición.

➤ Lista de Miembros.

➤ Deivis González – 60% de trabajo en el Sprint.

➤ Alejandro León – 20% de trabajo en los Sprint.

➤ Yaima Toledo – 20% de trabajo en los Sprint.

➤ Un lugar y momento definidos para el Scrum Diario.

Laboratorio C de Informática – 9.00 a.m.

➤ Historias divididas en tareas.

Gestionar servidor ToD.	Crear interfaz gráfica.
	Habilitar/Deshabilitar y observar servidor ToD.
	Validar mensajes de salidas.
Visualizar equipos DOCSIS	Crear interfaz gráfica.
	Mostrar equipos DOCSIS según su ubicación.

Tabla 18: Historias divididas en tareas Sprint 5

CAPÍTULO 2. DESCRIPCIÓN Y CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA.

2.5 Diagrama de Caso de uso

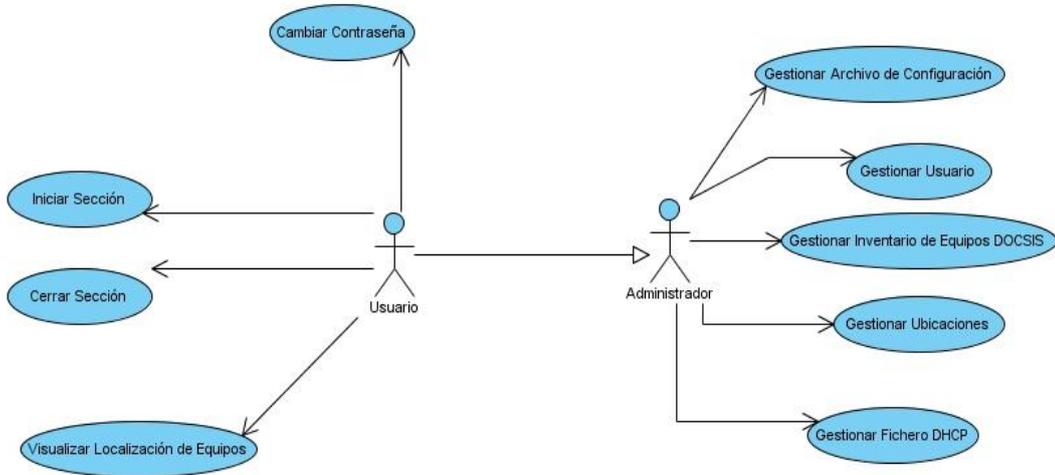


Figura 6: Diagrama de Casos de Usos

2.5.1 Descripción de los casos de uso

No.	Caso de Uso	Descripción	Prototipo
1	Iniciar Sesión	Anexo A.1	Anexo B.1
2	Cerrar Sesión	Anexo A.2	Anexo B.2
3	Cambiar Contraseña	Anexo A.3	Anexo B.3
4	Visualizar Localización de Equipos	Anexo A.4	Anexo B.4
5	Gestionar ubicaciones	Anexo A.5	Anexo B.5
6	Gestionar archivo de configuración	Anexo A.6	Anexo B.6, B.7, B.8, B.9, B.10, B.11, B.12, B.13, B.14
7	Gestionar Usuario	Anexo A.7	Anexo B.15
8	Gestionar inventario de equipos DOCSIS	Anexo A.8	Anexo B.16
9	Gestionar fichero DHCP	Anexo A.9	Anexo B.16, B.18

Tabla 19: Descripción de los Casos de Uso

CAPÍTULO 2. DESCRIPCIÓN Y CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA.

2.6 Diagramas

Modelo lógico de la base de datos

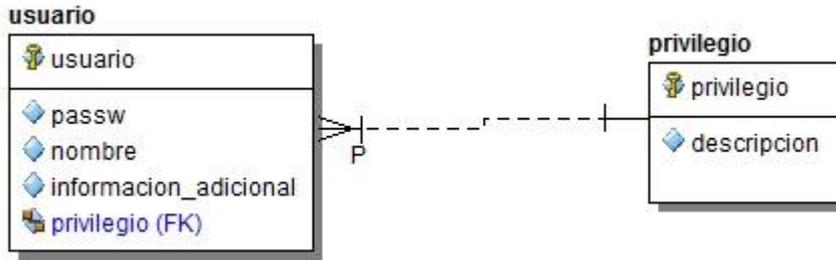


Figura 7: Modelo lógico de la base de datos Esquema Usuarios

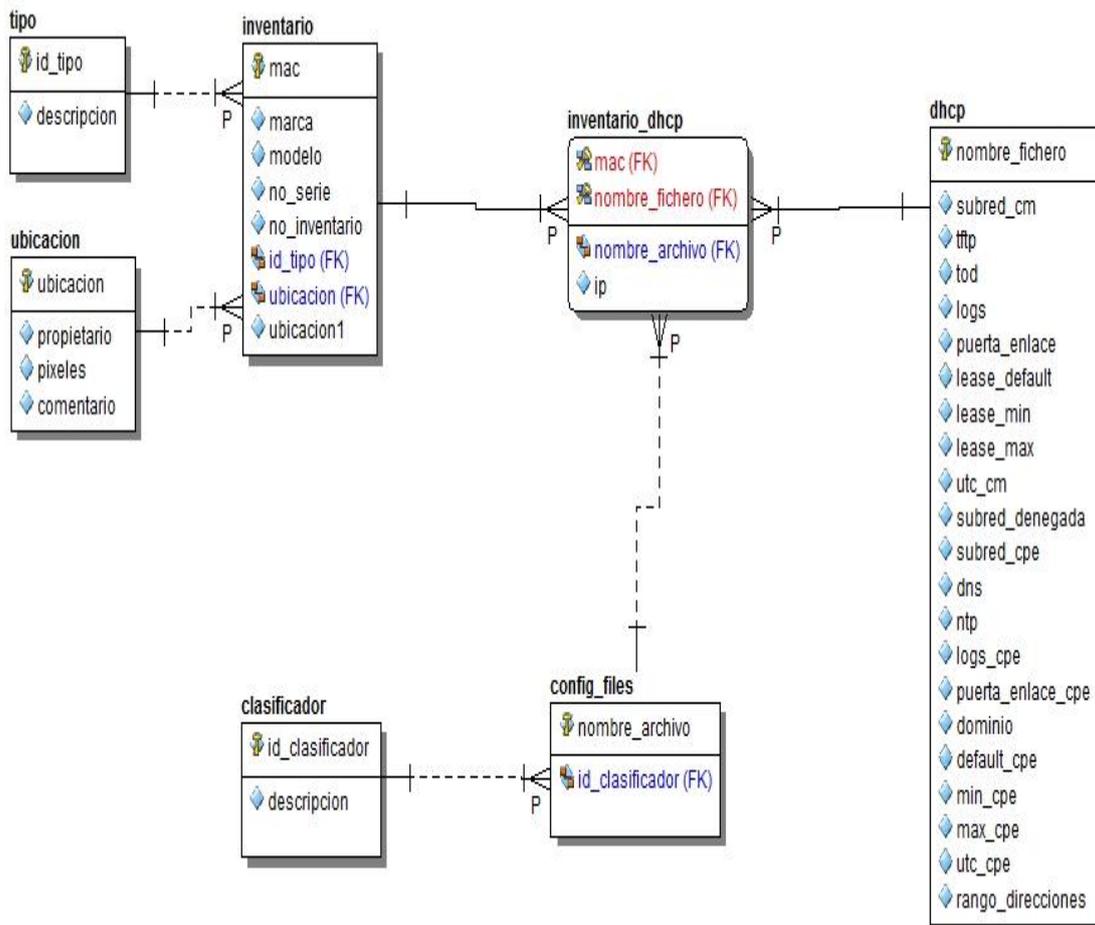


Figura 8: Modelo lógico de la base de datos Esquema Sistema

CAPÍTULO 2. DESCRIPCIÓN Y CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA.

Modelo físico de la base de datos

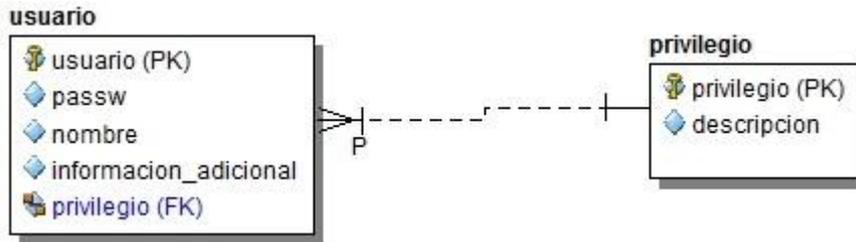


Figura 9: Modelo físico de la base de datos Esquema Usuarios

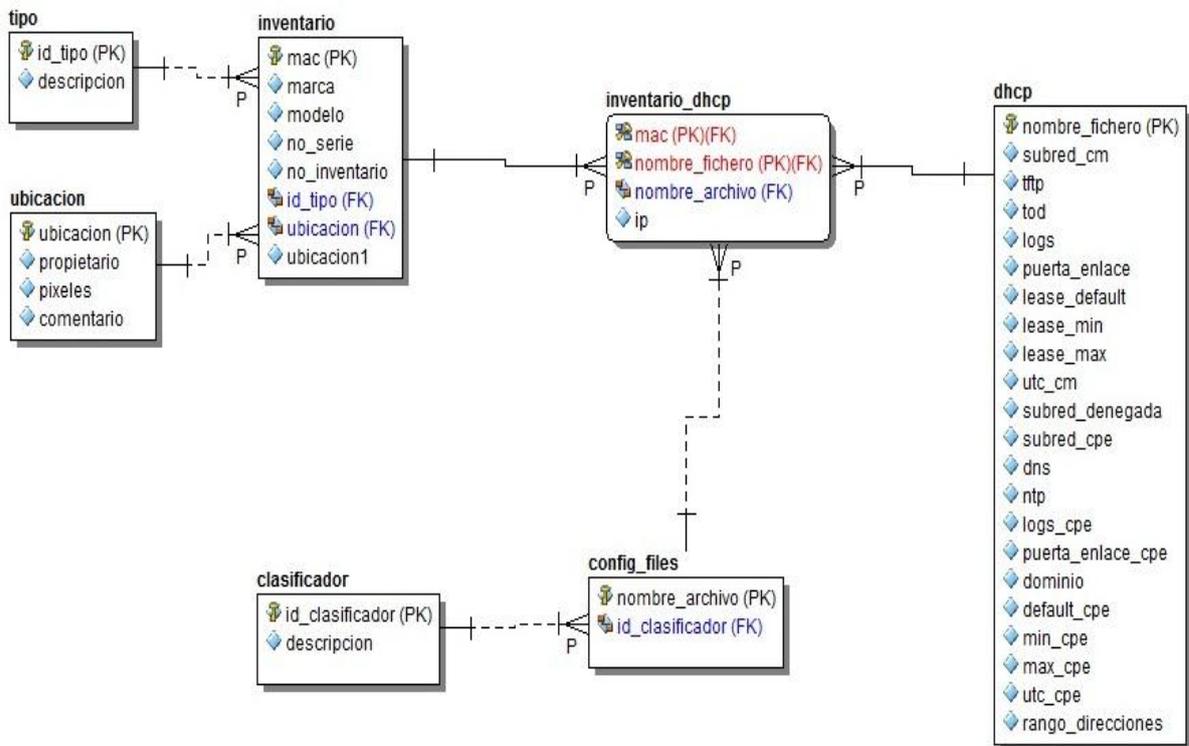


Figura 10: Modelo físico de la base de datos Esquema Sistema.

2.7 Principios del diseño del sistema.

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.

CAPÍTULO 2. DESCRIPCIÓN Y CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA.

2.7.1 Diseño de la interfaz de entrada, salida y menús del sistema.

La interfaz de entrada/salida y menús fue diseñada como se concibió íntegramente, aprovechando las potencialidades que ofrece Netbeans. Predomina el color gris oscuros en los formularios. La carga visual se distribuye de manera cómoda evitando acumulaciones engorrosas y cumpliendo con la regla de distribución de la atención: de izquierda a derecha y de arriba hacia abajo. El sistema cuenta con un diseño único, el cual lo integran un menú en la parte superior que permite al usuario el acceso a todas las funcionalidades en todo momento, en el medio dos pestañas para entrar al módulo de aprovisionamiento o al de gestión y cuando se accede a alguna funcionalidad se muestran los formularios sobre la principal siguiendo el mismo principio de diseño. La entrada de información por parte de los usuarios se realiza a través de los componentes del formulario.

2.7.2 Tratamiento de excepciones.

El sistema está diseñado e implementado de forma tal, que las posibilidades de introducir información errónea por parte del usuario sean mínimas, pues, aunque en muchas ocasiones el usuario teclea datos y en otras selecciona elementos de la pantalla, se mantiene un nivel de validación de la información a través de las técnicas de validación implementadas en Java y en caso de errores se comunica el error cometido a través de mensajes de error. Estos se muestran en un lenguaje de fácil comprensión para los usuarios.

2.7.3 Estándares de Codificación.

Para llevar a cabo una buena práctica de la ingeniería de software y el buen entendimiento del código, es necesario el uso de un estilo de código de manera que sea fácil de entender y que reduzca el tiempo y esfuerzo en el instante de realizar alguna modificación al mismo. Para ello se utilizaron en todo momento nombres descriptivos y comentarios informativos. Los nombres de las variables, métodos, consultas y objetos son cortos, claros, y describen su propósito, en idioma español.

CAPÍTULO 2. DESCRIPCIÓN Y CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA.

2.8 Conclusiones del Capítulo.

En este capítulo, tomando como guía la metodología de desarrollo SCRUM se realizó la pila del producto donde se definieron los requisitos funcionales del sistema, las pilas 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 y los requisitos no funcionales del sistema. También los diagramas del modelo físico y modelo lógico de datos de los que se obtuvo la representación de las relaciones existentes entre clases.

CAPÍTULO 3. ESTUDIO DE FACTIBILIDAD Y VALIDACIÓN DEL SISTEMA.

3.1 Introducción.

En este capítulo se presenta el estudio de factibilidad del sistema ofreciendo una descripción de la planificación del sistema. Se estiman el esfuerzo humano y el tiempo de desarrollo que se requieren para su elaboración, así como los costos por concepto de salario del mismo y los beneficios tangibles e intangibles que reporta. Se realiza además el análisis entre los costos y los beneficios para concluir si es o no factible su desarrollo, empleándose para ello el análisis de planificación por casos de uso. Se realiza además una validación del producto mediante una prueba de hipótesis.

3.2 Factibilidad.

Existe una variante para la estimación del esfuerzo en proyectos basados en Casos de Uso denominada Análisis de Puntos de Casos de Uso. Esta técnica permite cuantificar el tiempo de desarrollo de un proyecto, independientemente del lenguaje de programación, las metodologías, plataformas y/o tecnologías utilizadas, pero si teniendo en cuenta ciertos factores y su influencia en el proyecto [31].

3.2.1 Cálculo de Puntos por Casos de Uso sin Ajustar.

El primer paso para la estimación consiste en el cálculo de los Puntos de Casos de Uso sin ajustar. Este valor, se calcula a partir de la siguiente ecuación:

$$UUCP=UAW+UUCW$$

Dónde:

UUCP: Puntos de Casos de Uso sin ajustar.

UAW: Factor de Peso de los Actores sin ajustar.

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

CAPÍTULO 3. ESTUDIO DE FACTIBILIDAD Y VALIDACIÓN DEL SISTEMA.

3.2.2 Factor de Peso de los Actores sin ajustar (UAW).

Este valor se calcula mediante un análisis de la cantidad de Actores presentes en el sistema y la complejidad de cada uno de ellos. La complejidad de los Actores se establece teniendo en cuenta en primer lugar si se trata de una persona o de otro sistema, y en segundo lugar, la forma en la que el actor interactúa con el sistema.

Los criterios se muestran en la siguiente tabla:

Tipo de Actor	Descripción	Factor de Peso
Simple	Otro sistema que interactúa con el sistema a desarrollar mediante una interfaz de programación.	1
Medio	Otro sistema que interactúa con el sistema a desarrollar mediante un protocolo o una interfaz basada en texto.	2
Complejo	Una persona que interactúa con el sistema mediante una interfaz gráfica.	3

Tabla 20: Criterios del factor de peso de los actores sin ajustar.

Actor	Tipo de Actor
Administrador	Complejo
Supervisor	Complejo

Tabla 21: Clasificación de los actores del sistema.

Como se muestra en la tabla anterior existen en el sistema a desarrollar 2 actores de tipo complejo, ya que interactúan con el sistema mediante una interfaz gráfica.

Multiplicando la cantidad de actores de cada tipo por el peso correspondiente se obtiene que:

$$UAW = 2 * 3$$

$$UAW = 6$$

CAPÍTULO 3. ESTUDIO DE FACTIBILIDAD Y VALIDACIÓN DEL SISTEMA.

3.2.3 Factor de Peso de los Casos de Uso sin Ajustar.

Este valor se calcula mediante un análisis de la cantidad de casos de uso presentes en el sistema y la complejidad de cada uno de ellos. La complejidad de los casos de uso se establece teniendo en cuenta la cantidad de transacciones efectuadas en el mismo, donde una transacción se entiende como una secuencia de actividades atómica, es decir, se efectúa la secuencia de actividades completa, o no se efectúa ninguna de las actividades de la secuencia. Los criterios se muestran en la siguiente tabla:

Tipo de Caso de Uso	Descripción	Factor de Peso
Simple	El caso de uso contiene de 1 a 3 transacciones	5
Medio	El caso de uso contiene de 4 a 7 transacciones	10
Complejo	El caso de uso contiene más de 8 transacciones	15

Tabla 22: Criterios del factor de peso de los casos de uso sin ajustar.

A continuación se muestra la clasificación de los casos de uso del sistema:

Casos de Uso	Clasificación
Autenticar Usuario	Simple
Cambiar Contraseña	Simple
Gestionar Archivo de Configuración	Complejo
Gestionar Usuarios	Simple
Visualizar Localización de los Equipos	Medio

CAPÍTULO 3. ESTUDIO DE FACTIBILIDAD Y VALIDACIÓN DEL SISTEMA.

Gestionar Servicios TFTP	Simple
Gestionar Servicios de ToD	Simple
Modificar Fichero DHCP	Medio
Gestionar Inventario de Equipos DOCSIS	Medio

Tabla 23: Clasificador de los casos de uso del sistema.

En la tabla antes mostrada se tiene 5 casos de uso de clasificación simple, 3 casos de uso de clasificación medio y 1 caso de uso de clasificación complejo, por lo que se les aplican como factor de peso 5, 10 y 15 respectivamente.

Por tanto:

$$\mathbf{UUCW} = 5 * 5 + 3 * 10 + 1 * 15 = 25 + 30 + 15$$

$$\mathbf{UUCW} = 70$$

Como ya se dispone de los valores de factor de peso de actores y casos de uso sin ajustar es posible obtener el valor de los puntos de caso de uso sin ajustar es:

$$\mathbf{UUCP} = \mathbf{UAW} + \mathbf{UUCW}$$

$$\mathbf{UUCP} = 6 + 70$$

$$\mathbf{UUCP} = 76$$

3.2.4 Cálculo de Puntos de Casos de Uso ajustados.

Una vez que se tienen los Puntos de Casos de Uso sin ajustar, se debe ajustar éste valor mediante la siguiente ecuación:

$$\mathbf{UCP} = \mathbf{UUCP} \times \mathbf{TCF} \times \mathbf{EF}$$

Dónde:

UCP: Puntos de Casos de Uso ajustados

UUCP: Puntos de Casos de Uso sin ajustar

TCF: Factor de complejidad técnica

EF: Factor de ambiente

CAPÍTULO 3. ESTUDIO DE FACTIBILIDAD Y VALIDACIÓN DEL SISTEMA.

Es necesario calcular los valores de **TCF** y **EF**.

3.2.5 Factor de complejidad técnica (TCF).

Este coeficiente se calcula mediante la cuantificación de un conjunto de factores que determinan la complejidad técnica del sistema. Cada uno de los factores se cuantifica con un valor de 0 a 5, donde 0 significa un aporte irrelevante y 5 un aporte muy importante.

En la siguiente tabla se muestra el significado, el peso, el valor asignado y el total:

Factor	Descripción	Peso	Valor	Comentarios	Σ (Peso(i)* Valor(i))
T1	Sistema Distribuido	2	3	Aplicación Desktop	6
T2	Objetivos de performance o tiempo de respuesta	1	4	Se requiere que el sistema tenga un buen rendimiento.	4
T3	Eficiencia del Usuario Final	1	5	El resultado final debe ser eficiente	5
T4	Procesamiento Interno Complejo	1	5	Existen cálculos complejos.	5
T5	El código debe ser reutilizable	1	3	El código no es en su totalidad reutilizable.	3
T6	Facilidad de Instalación	0.5	4	El sistema es fácil de instalar.	2
T7	Facilidad de Uso	0.5	4	Es de fácil uso.	2
T8	Portabilidad	2	3	Se requiere que el sistema sea portable.	6
T9	Facilidad de Cambio	1	3	Fácil respuesta a cambios.	3
T10	Incluye objetivos	1	3	Se requiere	3

CAPÍTULO 3. ESTUDIO DE FACTIBILIDAD Y VALIDACIÓN DEL SISTEMA.

	especiales de seguridad			seguridad normal.	
T11	Se requiere facilidades especiales de entrenamiento de usuarios	1	3	Facilidad normal de uso.	3

Tabla 24: Significado y pesos de los TCF.

El Factor de complejidad técnica se calcula mediante la siguiente ecuación:

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

$$\text{TCF} = 0.6 + 0.01 * (6+4+5+5+3+2+2+6+3+3+3)$$

$$\text{TCF} = 0.6 + 0.01 * 42*0.01$$

$$\text{TCF} = 1.02$$

3.2.6 Factor ambiente (EF).

Las habilidades y el entrenamiento del grupo involucrado en el desarrollo tienen un gran impacto en las estimaciones de tiempo. Estos factores son los que se contemplan en el cálculo del Factor de ambiente. El cálculo del mismo es similar al cálculo del Factor de complejidad técnica, es decir, se trata de un conjunto de factores que se cuantifican con valores de 0 a 5.

Factor	Descripción	Peso	Valor	Comentarios	$\Sigma (\text{Peso}(i) * \text{Valor}(i))$
E1	Familiaridad con el modelo de proyecto utilizado.	1.5	3	Existe cierta familiaridad con el modelo de proyecto.	4.5
E2	Experiencia con la aplicación	0.5	3	Hay poca experiencia en la aplicación.	1.5
E3	Experiencia en la orientación a objeto	1	4	La mayoría del grupo ha	4

CAPÍTULO 3. ESTUDIO DE FACTIBILIDAD Y VALIDACIÓN DEL SISTEMA.

				programado Orientado a Objetos.	
E4	Motivación	1	5	Existe gran motivación por el proyecto.	5
E5	Estabilidad de los requerimientos	2	3	No se esperan cambios	6
E6	Personal part-time	-1	0	Todos trabajan a tiempo completo.	0
E7	Dificultad del lenguaje de programación	-1	3	No existe gran dificultad en el lenguaje de programación utilizado, el cual fue Java.	-3

Tabla 25: Significado y peso de las habilidades del grupo.

El Factor de ambiente se calcula mediante la siguiente ecuación:

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

$$EF = 1.4 - 0.03 * (4.5 + 1.5 + 4 + 5 + 6 + 0 - 3)$$

$$EF = 1.4 - 0.03 * (18)$$

$$EF = 0.86$$

Con el cálculo de estos valores, es posible sustituir en la ecuación inicial y obtener el valor de los puntos de caso de uso ajustado.

De esta forma:

$$UCP = UUCP \times TCF \times EF$$

$$UCP = 76 * 1.02 * 0.86$$

$$UCP = 66.6672$$

3.2.7 Estimación del esfuerzo.

$$E = UCP * CF$$

CAPÍTULO 3. ESTUDIO DE FACTIBILIDAD Y VALIDACIÓN DEL SISTEMA.

Dónde:

E: esfuerzo estimado en horas-hombre.

UCP: Puntos de Casos de Uso ajustados.

CF: Factor de conversión.

Para calcular Factor de Conversión (CF):

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

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

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

En este proyecto si se analizan los valores tabulados anteriormente, es posible percatarse que el total es menor que 2, por lo que se utiliza el factor de conversión 20 horas-hombre/Punto de Casos de Uso.

De esta forma se obtiene que el esfuerzo necesario para desarrollar los casos de uso del sistema es igual a:

$$E = UCP \times CF$$

$$E = 66.6672 * 20$$

$$E = 1333.344 \text{ Horas-Hombre}$$

Duración:

Trabajando 25 días al mes y 9 horas diarias como promedio, se tiene que:

$$\text{Duración (días)} = \text{Total de Horas / Hombre entre 9 horas al día} = 1333.344 / 9 \\ = 148.15 \text{ días.}$$

$$\text{Duración (meses)} = \text{Total de días / 25 días por mes} = 148.15 / 25 = \approx 6 \text{ meses.}$$

Se debe tener en cuenta que éste método proporciona una estimación del esfuerzo en horas-hombre contemplando sólo el desarrollo de la funcionalidad especificada en los casos de uso. Por lo que para obtener una estimación más completa de la duración total del proyecto, hay que agregar a la estimación del esfuerzo obtenida, las estimaciones de esfuerzo de las demás actividades relacionadas con el desarrollo de software. Existe un criterio que estadísticamente se considera aceptable, que distribuye el esfuerzo de las diferentes actividades dentro del desarrollo de un proyecto según la estimación

CAPÍTULO 3. ESTUDIO DE FACTIBILIDAD Y VALIDACIÓN DEL SISTEMA.

que se muestra en la tabla siguiente, a la que también se le ha agregado el cálculo del valor del esfuerzo para el sistema de esta investigación:

Actividad	Porcentaje	Valor
Análisis	10%	835.28
Diseño	20%	1670.56
Programación	40%	3341.12
Prueba	15	1252.92
Sobrecarga	15%	1252.92
Total de horas	100%	8352.8

Tabla 26: Criterios de distribución de esfuerzos.

3.2.8 Cálculo de costos:

Tomando como salario promedio mensual \$300.00.

Costo = 6 meses * \$300.00 = \$1800.00

Costo = \$1800.00 MN

Costo = \$72.00 CUC

3.2.9 Beneficios tangibles e intangibles.

Todo producto en su etapa de desarrollo trae consigo un costo. La decisión de emprender el proyecto depende en cierta medida de los beneficios económicos o sociales, o ambos, que se obtengan con el futuro uso de sus resultados.

El costo de la realización de este producto informático se estimó en el orden de los \$ 1800.00 en moneda nacional, aplicando las tasas de cambios del peso convertible cubano (CUC), que 1.00 CUC equivale a 25.00 pesos cubanos se tiene que la aplicación tuvo un costo de alrededor de los \$ 72.00 CUC. Es interesante resaltar que el costo estimado está muy por debajo del precio en el mercado internacional de los sistemas similares al sistema informático propuesto.

Los beneficios obtenidos con el desarrollo del software permiten agilizar los procesos de la conformación de archivos de configuración para abonados y de la configuración de los servicios DHCP y TFTP.

CAPÍTULO 3. ESTUDIO DE FACTIBILIDAD Y VALIDACIÓN DEL SISTEMA.

3.2.10 Análisis de los costos y beneficios.

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

El sistema, resultado del presente trabajo de diploma, no implica costo alguno para la empresa Telecable, siendo de gran aporte para el capital monetario de la misma, el cual permite el futuro desarrollo de servicios como internet, video a demanda entre otros. Con el propósito de mejorar las actividades y ampliar la cartera de ofertas en este polo turístico.

El desarrollo de este sistema informático, representa un costo total de \$1800.00 MN (\$ 72.00 CUC) y se puede llevar a cabo por una sola persona en un plazo de 6 meses.

3.3 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 [32].

Para realizar la validación del sistema informático propuesto se tomaron 15 observaciones del tiempo en minutos en que demoraba el proceso de aprovisionamiento a una red DOCSIS antes y después del sistema informático. Se observó que el procesamiento antes del software demoraba como promedio 17 minutos y después solo 9 minutos, 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 medias antes y después del sistema.

CAPÍTULO 3. ESTUDIO DE FACTIBILIDAD Y VALIDACIÓN DEL SISTEMA.

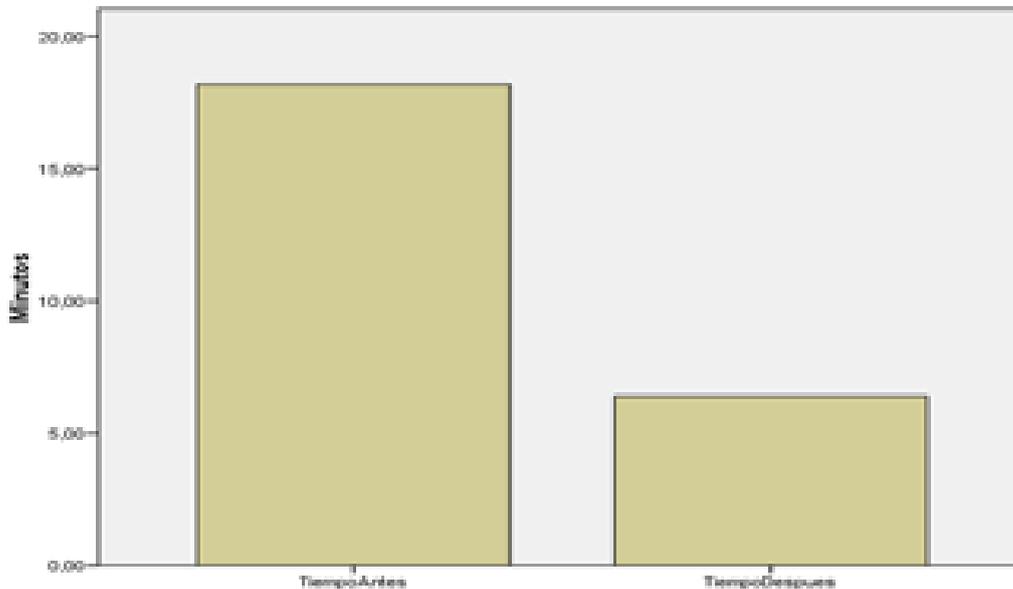


Figura 11: Comparación del tiempo promedio de los procesos antes y después del sistema.

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 sigue una distribución normal contra la hipótesis alternativa en que se considera que la variable no sigue una distribución normal. Tomando como referencia un nivel de significación del 5%, si este es mayor que la significación asintótica, entonces rechazamos H_0 , de lo contrario aceptamos. Utilizando un nivel de significación de 0,05 al comparar con la significación asintótica de los estadísticos calculados (0.452 y 0,331) puede concluirse que no se rechaza la hipótesis nula, demostrando que ambas variables siguen una distribución normal, por tanto al cumplirse este supuesto puede realizarse la Prueba T. (Ver Anexo F1).

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

CAPÍTULO 3. ESTUDIO DE FACTIBILIDAD Y VALIDACIÓN DEL SISTEMA.

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.(ver Anexo F2).

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

3.4 Conclusiones del Capítulo.

La realización del estudio de factibilidad del producto informático proyectó una cantidad significativa de beneficios tangibles e intangibles. El sistema propuesto contribuye de forma positiva en el proceso de Aprovechamiento a una red DOCSIS de la empresa Telecable Internacional Cayos Santa María.

Una vez concluido el estudio de factibilidad del sistema, se estima un tiempo de 6 meses para su construcción por un hombre y su costo asciende a \$1800.00.

El desarrollo de la validación del sistema mostró resultados favorables a partir de la prueba de hipótesis donde el ahorro de tiempo y la confiabilidad son los mejores beneficios que aporta el sistema.

CONCLUSIONES

A partir de los objetivos planteados con anterioridad se arriba a las siguientes conclusiones:

- Para la confección del sistema propuesto se realizó un análisis de la forma en que se realiza el proceso de aprovisionamiento a una red DOCSIS, las principales características y funcionamiento del estándar, lo cual permitió lograr un mejor entendimiento del proceso que se encarga Telecable internacional para brindar los servicios con la mejor calidad posible.
- Se analizaron los protocolos de administración de redes vinculados con el problema y sus implementaciones por los diferentes lenguajes de programación, este análisis permitió escoger el lenguaje con el que se implementó el sistema y se logró desarrollar al máximo las funcionalidades que eran necesarias para el funcionamiento del mismo.
- Se implementó un sistema de aprovisionamiento para una red DOCSIS logrando así su correcto funcionamiento y abasteciendo a la red con la información necesaria para su correcto funcionamiento.
- Se probaron los resultados en un entorno real comprobando que el sistema funciona correctamente.

RECOMENDACIONES

A pesar de que la investigación realizada cumplió con los objetivos trazados, se recomienda:

- Poner en Prueba todas las funcionalidades del sistema propuesto durante un período de tiempo, permitiendo esto comprobar dichas funcionalidades de forma práctica y poder entonces detectar posibles mejoras a realizar en el futuro.
- Extender el uso del sistema en lugares donde exista el equipamiento DOCSIS, para el aprovisionamiento de la misma.
- Continuar con el estudio de los servicios que brinda la empresa Telecable con el objetivo de añadirle nuevos módulos al sistema.

REFERENCIAS BIBLIOGRÁFICAS

- [1] «A Tutorial on DOCSIS: Protocol and Performance Models», University of Bradford, UK & Clemson University, USA, 2006.
- [2] GREEN, J. H, *Access Technologies: DSL and Cable*, Inc. USA: McGraw-Hill Companies, 2002.
- [3] «Introduction to Cable Television».
- [4] CABLE TELEVISION LABORATORIES, I, «MAC and Upper Layer Protocols Interface Specification. Data Over Cable Service Interface Specifications DOCSIS 3.0», USA, 2011.
- [5] CABLE EUROPE LABS, *Overview of Architecture, Technical Features and Services of Integrated Broadband and Cable TV Networks*. Bruselas, Bélgica: Cable Europe®, 2009.
- [6] PICI, L. G, *Introduction to Cable Television*. USA: 2007.
- [7] TOOLEY, M y BOWMAN, D, «An overview of the DOCSIS (Cable Internet) Platform», Sandvine® Intelligent Broadband Networks, 2010.
- [8] CABLE TELEVISION LABORATORIES, I, «Physical Layer Specification. Data Over Cable Service Interface Specifications DOCSIS 3.0», USA, 2010.
- [9] CARCAMO, C, «Instalación CMTS», 2009. [Online]. Available: www.tripleplay.mx/support/instalacion-cmts.pdf.
- [10] VOLPE, B, «DOCSIS Tutorial Series». bradyvolpe.com, 2009.
- [11] 10. DOWNEY, J. J, *DOCSIS 3.0 Overview*, Inc. Cisco Systems, 2006.
- [12] VECIMA NETWORKS, *M-CMTS & DOCSIS 3.0 Standards Overview*. Vecima Networks, 2008.
- [13] Amaro Calderón, Sarah Dámaris Valverde Rebaza. Jorge Carlos, «Metodologías Ágiles». 2007.
- [14] «Metodologías Tradicionales». [Online]. Available: <http://www.eumed.net/Metodologiastradicionales.htm>.
- [15] Amaro Calderón, Sarah Dámaris Valverde Rebaza. Jorge Carlos, «Metodologías Ágiles». 2007.
- [16] X. F. Grau, «Desarrollo orientado a objetos con UML», mar-2008. [Online]. Available: <http://www.clikear.com/manuales/uml/introduccion.asp>.
- [17] Sun Microsystems, Inc, «El lenguaje de Programación Java™».
- [18] Ronald Peñarrieta, «Java y NetBeans». 2011.
- [19] T. Lockhart, «Tutorial de PostgreSQL».
- [20] «Lenguaje SQL», 02-jul-2009. [Online]. Available: <http://www.arsys.es/soporte/programacion>.
- [21] «Sql», 02-jul-2009. [Online]. Available: <http://www.arsys.es/soporte/programacion>.
- [22] Walter Celiano Tituaña y Edwin JESus Torres, «NetBeans». 2009.
- [23] «Visual Paradigm - EcuRed», 07-feb-2013. [Online]. Available: http://www.ecured.cu/index.php/Visual_Paradigm. [Accessed: 07-feb-2013].
- [24] «Visual Paradigm». [Online]. Available: http://www.ecured.cu/index.php/Visual_Paradigm. [Accessed: 09-abr-2013].
- [25] «Scrum y XP desde las trincheras».

BIBLIOGRAFÍA

- [1] N. Shah y D. Kouvatsos, «A Tutorial on DOCSIS: Protocol and Performance Models».
- [2] «A Tutorial on DOCSIS: Protocol and Performance Models», University of Bradford, UK & Clemson University, USA, 2006.
- [3] GREEN, J. H, *Access Technologies: DSL and Cable*, Inc. USA: McGraw-Hill Companies, 2002.
- [4] A. Cockburn, «Agile Software Development. Highsmith Series. » .
- [5] M. Shaw, «An Introduction to Software Architecture». ene-1994.
- [6] TOOLEY, M y BOWMAN, D, «An overview of the DOCSIS (Cable Internet) Platform», Sandvine® Intelligent Broadband Networks, 2010.
- [7] O. Muñoz, «Arquitectura de aplicaciones Web», presented at the Conferencia de Seminarios Especiales I, 2004.
- [8] «Arquitectura de Software». [Online]. Available: http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/hernandez_j_pg/capitulo2.pdf
- [9] 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].
- [10] «Delphi», 06-feb-2012. [Online]. Available: <http://www.larevistainformatica.com/Delphi.htm>.
- [11] X. F. Grau, «Desarrollo orientado a objetos con UML», mar-2008. [Online]. Available: <http://www.clikear.com/manuales/uml/introduccion.asp>.
- [12] 10. DOWNEY, J. J, *DOCSIS 3.0 Overview*, Inc. Cisco Systems, 2006.
- [13] VOLPE, B, «DOCSIS Tutorial Series». bradyvolpe.com, 2009.
- [14] Sun Microsystems, Inc, «El lenguaje de Programación Java™».
- [15] «Enciclopedia Universal Ilustrada Europeo Americana.», *Enciclopedia Universal Ilustrada Europeo Americana*. .
- [16] Center for History and New Media, «Guía rápida». [Online]. Available: http://zotero.org/support/quick_start_guide.
- [17] «HFC Resumen».
- [18] Pete Deemer, «Información Básica de Scrum». 2009.
- [19] CARCAMO, C, «Instalación CMTS», 2009. [Online]. Available: www.tripleplay.mx/support/instalacion-cmts.pdf.
- [20] R. C. Carrasco, «Introducción al diseño y a la programación orientada a objetos», mar-2009. [Online]. Available: <http://www.nielsoft.com/Seminario/3capas/introduccion.ppt>.
- [21] L. Toledo, «Introducción al Lenguaje SQL», presented at the Conferencia de Sistemas de Bases de Datos, 2005.
- [22] PICI, L. G, *Introduction to Cable Television*. USA: , 2007.
- [23] «Introduction to Cable Television».
- [24] J. Sánchez, «Java 2 incluye swing, threads, programación en red, jdbc y jsp». 2003.
- [25] «Java 2 Manual FV».
- [26] «Java Persistent Objects - JPA Tutorial», *Persistencia de objetos con java*, 21-oct-2011. [Online]. Available: http://www.jpox.org/docs/1_2/jpa_orm/index.html. [Accessed: 21-oct-2011].
- [27] Ronald Peñarrieta, «Java y NetBeans». 2011.

-
- [28] Ronaldo Peñerrieta Capriloro, *Java y NetBeans*. 2009.
- [29] J. Sánchez, «Java2». 2003.
- [30] A. Z. Gorostiza, «La Disciplina de Arquitectura». [Online]. Available: <http://profesores.fi-b.unam.mx/adanzg/SwEng/Presentaciones/DisciplinaArquitectura.pdf>.
- [31] «leer un excel con java - Desarrollo de sitios web», *leer un excel con java*, 05-nov-2011. [Online]. Available: <http://www.todoexpertos.com/categorias/tecnologia-e-internet/desarrollo-de-sitios-web/expertos/joraanma>. [Accessed: 05-nov-2011].
- [32] «Lenguaje SQL», 02-jul-2009. [Online]. Available: <http://www.arsys.es/soporte/programacion>.
- [33] CABLE TELEVISION LABORATORIES, I, «MAC and Upper Layer Protocols Interface Specification. Data Over Cable Service Interface Specifications DOCSIS 3.0», USA, 2011.
- [34] VECIMA NETWORKS, *M-CMTS & DOCSIS 3.0 Standards Overview*. Vecima Networks, 2008.
- [35] Amaro Calderón, Sarah Dámaris Valverde Rebaza. Jorge Carlos, «Metodologías Ágiles». 2007.
- [36] Amaro Calderón, Sarah Dámaris Valverde Rebaza. Jorge Carlos, «Metodologías Ágiles». 2007.
- [37] «Metodologías Tradicionales». [Online]. Available: <http://www.eumed.net/Metodologíastradicionales.htm>.
- [38] «MySql Características», 16-ene-2012. [Online]. Available: <http://dev.mysql.com/tech-resources/articles/dispelling-the-myths.html>.
- [39] Walter Celiano Tituaña y Edwin JESus Torres, «NetBeans». 2009.
- [40] «NetBeans - EcuRed», 07-feb-2013. [Online]. Available: <http://www.ecured.cu/index.php/NetBeans>. [Accessed: 07-feb-2013].
- [41] «Oracle», 11-nov-2011. [Online]. Available: <http://www.desarrolloweb.com/articulos/840.php>.
- [42] «Origen y formación del petróleo», feb-2008. [Online]. Available: <http://elpetroleo.aop.es/Tema1/Index1.asp>.
- [43] CABLE EUROPE LABS, *Overview of Architecture, Technical Features and Services of Integrated Broadband and Cable TV Networks*. Bruselas, Bélgica: Cable Europe®, 2009.
- [44] «Pascal», 06-feb-2012. [Online]. Available: <http://www.slideshare.net/otaru/lenguaje-pascal>.
- [45] CABLE TELEVISION LABORATORIES, I, «Physical Layer Specification. Data Over Cable Service Interface Specifications DOCSIS 3.0», USA, 2010.
- [46] «PostgreSQL», 19-oct-2011. [Online]. Available: <http://grupove.org.ve/postgresql>.
- [47] «Problema al cargar la página», 07-abr-2013. [Online]. Available: <http://www.arsys.es/soporte/programacion>. [Accessed: 07-abr-2013].
- [48] «Procesando XML en Java [Xerces]», *Procesando XML en Java [Xerces]*, 21-oct-2011. [Online]. Available: <http://casidiablo.net/xml-java/>. [Accessed: 21-oct-2011].
- [49] N. S. Kouvatso, «Protocol and Performance Models», *University of Bradford, UK*.
- [50] C. Ruata y J. Palacio, «Scrum Manager Proyectos». 2009.
- [51] «Scrum y XP desde las trincheras».
- [52] «Sprintometer», 17-ene-2012. [Online]. Available: <http://www.exactamente.com/2012/01/sprintometer/>.

-
- [53] «Sql», 02-jul-2009. [Online]. Available: <http://www.arsys.es/soporte/programacion>.
- [54] «SQL Server», 04-nov-2011. [Online]. Available: <http://www.arsys.es/soporte/programacion>.
- [55] «Swing Progress Bar,Java Progress Bar Example,How to Create Progressbar in Java», 05-nov-2011. [Online]. Available: <http://www.roseindia.net/java/example/java/swing/SwingProgressBar.java>. [Accessed: 05-nov-2011].
- [56] «Televisión por cable - EcuRed», 07-feb-2013. [Online]. Available: http://www.ecured.cu/index.php/Televisi%C3%B3n_por_cable. [Accessed: 07-feb-2013].
- [57] J. Tulach, «The Definitive Guide to NetBeans Platform». 2009.
- [58] ADER, J y CLOONAN, T, *Thomas Wiesel Partners DOCSIS® 3.0 Tutorial*, Inc. ARRIS Group, 2007.
- [59] M. Domínguez-Dorado, *Todo Programación*. Madrid: Iberprensa, 2005.
- [60] T. Lockhart, «Tutorial de PostgreSQL».
- [61] «Visual Basic», 06-feb-2012. [Online]. Available: http://www.adrformacion.com/curso/visualbasic/leccion1/introduccion_visual_basico.htm.
- [62] «Visual Paradigm - EcuRed», 07-feb-2013. [Online]. Available: http://www.ecured.cu/index.php/Visual_Paradigm. [Accessed: 07-feb-2013].

ANEXOS

Anexo. A - Descripción de los casos de uso del sistema.

Anexo A.1 – Descripción del caso de uso iniciar sesión.

Caso de Uso.	Autenticar usuario.
Actores	Usuario.
Propósito	Permitir el acceso a las funcionalidades del sistema.
Resumen	El caso de uso se inicia cuando el usuario desea ingresar al sistema. Para ello debe ingresar su identificador como usuario, y contraseña, luego de esto se chequea. Si los datos del usuario son correctos podrá acceder a las opciones del sistema que le corresponden según sus privilegios, en caso contrario se mostrará un mensaje de error, denegando el acceso. Terminando así el caso de uso.
Prototipos	B.1
Precondiciones	El usuario debe estar registrado en el sistema, teniendo un nombre de usuario y una contraseña.
Post- condiciones	El usuario tendrá acceso a las funcionalidades del sistema disponibles al su nivel de privilegio.

Anexo A.2 – Descripción del caso de uso cerrar sesión.

Caso de Uso.	Cerrar sesión.
Actores	Usuario.
Propósito	Permitir al usuario cerrar la sesión iniciada.
Resumen	El caso de uso inicia cuando el usuario desea cerrar su sesión y para ello selecciona en el menú la opción salir de la aplicación. El caso de uso concluye cuando el sistema muestra la pantalla de autenticar usuario.
Prototipos	B.2
Precondiciones	El usuario debe haberse autenticado previamente.
Post- condiciones	Se cierra la sesión iniciada y se muestra la pantalla de autenticar usuario.

Anexo A.3 – Descripción del caso de uso cambiar contraseña.

Caso de Uso.	Cambiar contraseña.
Actores	Usuario.
Propósito	Permitir al usuario cambiar la contraseña de acceso al sistema.
Resumen	El caso de uso inicia cuando el usuario desea cambiar su contraseña y para ello selecciona en el menú la opción cambiar contraseña, donde se muestra un formulario con los datos del usuario. Para ello debe ingresar su contraseña anterior y la nueva a sustituir, luego se chequea, si la contraseña anterior coincide con la almacenada en el sistema. Si coincide se modifica satisfactoriamente, en caso contrario se mostrará un mensaje de error, denegando la modificación. Culminando así el caso de uso.
Prototipos	B.3
Precondiciones	El usuario debe haberse autenticado previamente.
Post-condiciones	Se muestra la pantalla de principal del sistema.

Anexo A.4 – Descripción del caso de uso Visualizar localización de equipos.

Caso de Uso.	Visualizar localización de equipos.
Actores	Usuario.
Propósito	Permitir al usuario ver la localización de los equipos.
Resumen	El caso de uso inicia cuando el usuario inicia la sesión, donde se visualiza en la página principal la ubicación de todos los equipos, pudiendo saber dónde está ubicado cada cual, la cantidad que existe por ubicación, los que están conectados en online y los que están en espera.
Prototipos	B.4
Precondiciones	El usuario debe haberse autenticado previamente.
Post-condiciones	Se cierra la sesión iniciada y se muestra la pantalla de autenticar usuario.

Anexo A.5 – Descripción del caso de uso Gestionar ubicaciones.

Caso de Uso.	Gestionar ubicaciones.
Actores	Administrador.
Propósito	Permitir al administrador insertar, modificar o eliminar ubicaciones.
Resumen	El caso de uso inicia cuando el administrador desea insertar, modificar o eliminar una ubicación y selecciona en el menú la opción de gestionar ubicación donde se visualizan los equipos ya insertados y se brindan las opciones de agregar, modificar o eliminar. Si desea agregar una ubicación debe insertar los datos correspondientes a las ubicaciones, el sistema verifica que la ubicación a insertar no exista, de existir, el sistema muestra un mensaje de error informando que la ubicación ya existe, en caso de otro error, muestra un mensaje pidiendo al administrador que verifique los datos insertados. Si desea modificar debe seleccionar en la lista la ubicación donde modifica los datos deseados, el sistema verifica que la ubicación a modificar no exista, de existir, el sistema muestra un mensaje de error informando que la ubicación ya existe, en caso de otro error, muestra un mensaje pidiendo al administrador que verifique los datos modificados. Si desea eliminar selecciona en la lista la(s) ubicación(es) a eliminar y muestra un mensaje si desea eliminar la(s) ubicación(es) seleccionadas, en caso de aceptar elimina la(s) ubicación(es), sino no la elimina.
Prototipos	B.5
Precondiciones	El administrador debe haberse autenticado previamente.
Post-condiciones	Se actualizan la información de las ubicaciones en la base de datos.

Anexo A.6 – Descripción del caso de uso Gestionar archivo de configuración.

Caso de Uso.	Gestionar archivo de configuración.
Actores	Administrador.

Propósito	Permitir al administrador insertar, modificar o eliminar archivos de configuración.
Resumen	El caso de uso inicia cuando el administrador desea insertar, modificar o eliminar un archivo de configuración y selecciona en el menú la opción de gestionar archivos de configuración donde se muestra en una lista los archivos de configuración existentes y listando los datos principales del primero en la lista, se brindan las opciones de agregar, modificar o eliminar. Si desea agregar un archivo de configuración debe dar clic en el botón de insertar donde se muestran las interfaces con los datos necesarios para insertar un archivo de configuración y después de haber terminado de llenar los datos selecciona el botón de aplicar y crea el archivo de configuración mostrando la lista de los archivos con la información principal del archivo insertado, en caso de modificar seleccione el archivo en la lista y da clic en el botón editar, cambia los datos deseados da clic en el botón aplicar y se muestra la lista de los archivos con la información del modificado. En el caso de eliminar selecciona el/los archivo(s) en la lista y da clic en el botón eliminar y muestra un mensaje si desea eliminar los archivos seleccionados
Prototipos	B.6, B.7, B.8, B.9, B.10, B.11, B.12, B.13, B.14
Precondiciones	El administrador debe haberse autenticado previamente.
Post-condiciones	Se actualizan la información de los archivos de configuraciones en la base de datos.

Anexo A.7 – Descripción del caso de uso Gestionar Usuarios.

Caso de Uso.	Gestionar Usuarios.
Actores	Administrador.
Propósito	Permitir al administrador insertar, modificar o eliminar un usuario del sistema.
Resumen	El caso de uso se inicia cuando el administrador desea

	<p>insertar, modificar o eliminar usuarios y para ello selecciona la opción gestión usuario en el menú seguridad. Inicialmente se listan todo los usuarios existentes y se brindan las opciones de nuevo, eliminar y editar. Si desea agregar un nuevo usuario el administrador deberá insertar los datos correspondientes con los usuarios, en caso de ya existir ese usuario, muestra un mensaje avisando que el usuario ya existe, si la contraseña insertada no coincide con la contraseña verificar muestra un mensaje avisando que las contraseñas no coinciden. Si desea editar un usuario, se selecciona el mismo en la lista y se modifican los datos deseados, si desea editar la contraseña debe llenar los campos contraseña y continuar, sino se dejan los dos campos en blanco y el sistema no modifica la contraseña. En caso de eliminar selecciona en la lista el/los usuario(s) seleccionados, al ejecutar la opción eliminar muestra un mensaje avisando si desea eliminar el/los usuario(s) seleccionados, en caso de aceptar ejecuta la acción y elimina el/los usuario(s) seleccionados.</p>
Prototipos	B.15
Precondiciones	El administrador debe haberse autenticado previamente.
Post- condiciones	Se actualiza la información de los usuarios en la base de datos.

Anexo A.8 – Descripción del caso de uso Gestionar inventario de equipos DOCSIS.

Caso de Uso.	Gestionar inventario de equipos DOCSIS.
Actores	Administrador.
Propósito	Permitir al administrador insertar, modificar o eliminar los equipos DOCSIS.
Resumen	El caso de uso se inicia cuando el administrador desea insertar, modificar o eliminar equipos DOCSIS y para ello selecciona la opción dispositivos y se brindan las opciones

	de agregar, eliminar y modificar. Se muestra la lista de los equipos DOCSIS existentes, si desea agregar un nuevo equipo el administrador deberá insertar los datos correspondientes, cuando da en aceptar, el sistema verifica que el equipo no exista, en caso de ya existir envía un mensaje de error avisando que ya existe. Si lo que desea es eliminar o modificar un equipo DOCSIS, el administrador deberá seleccionar de la lista cual equipo es el que desea eliminar o modificar, si es eliminar el sistema le pedirá confirmar la eliminación. Si se trata de modificar un equipo DOCSIS, debe seleccionar en la lista el equipo que desea modificar y el administrador podrá cambiar los datos que desea modificar. El caso de uso termina con la actualización de los datos.
Prototipos	B.16
Precondiciones	El administrador debe haberse autenticado previamente.
Post- condiciones	Se actualiza la información de los equipo DOCSIS en la base de datos.

Anexo A.9 – Descripción del caso de uso Gestionar fichero DHCP.

Caso de Uso.	Gestionar fichero DHCP.
Actores	Administrador.
Propósito	Permitir al administrador agregar, modificar o eliminar los ficheros de DHCP.
Resumen	El caso de uso se inicia cuando el administrador desea insertar, modificar o eliminar ficheros DHCP y para ello selecciona la opción DHCP donde se muestra una interfaz con la información del fichero DHCP en control de perfiles muestra el nombre del fichero del cual se está mostrando los datos y un desplegable con los otros ficheros existentes en caso de ver los datos de algunos de ellos se selecciona y se da clic en el botón de Cargar, después de haber realizados los cambios deseados se da clic en el botón de Guardar y se

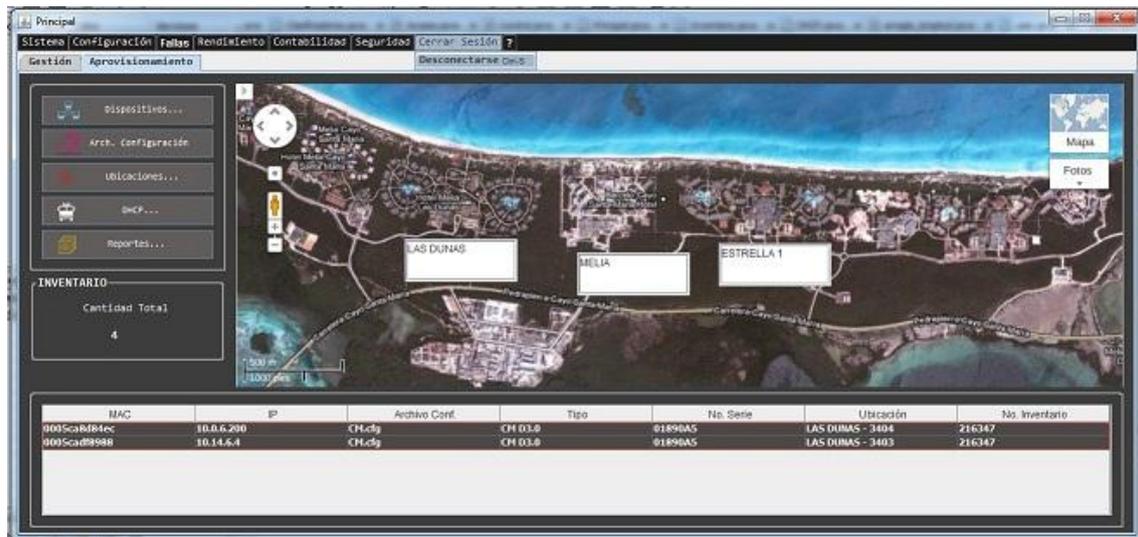
	guardan los datos, para insertar un cable modem a dicho fichero se da clic en el botón herramientas donde se muestra la interfaz de asignar direcciones de IP donde inicialmente se muestran dos listas una con los cables modem que no están relacionados con esa fichero y en la otra los que ya fueron relacionados con el fichero, en caso de relacionar un modem con ese fichero se selecciona el/los modem a relacionar se inserta un IP inicial el archivo de configuración de los mismos y lo asigna al fichero, el sistema se encarga de eliminarlos de la tabla de los no relacionados e insertarlos en la de los relacionados, se da clic en el botón guardar y se guardan los cambios realizados. Después de realizar cambios en los botones de control de servicios se da clic en el de reiniciar para que los modem reciban los cambios efectuados.
Prototipos	B.17, B.18
Precondiciones	El administrador debe haberse autenticado previamente.
Post-condiciones	Se actualiza la información correspondiente con los ficheros DHCP.

Anexo B. Prototipos.

Anexo B.1 – Prototipo del caso de uso iniciar sesión.



Anexo B.2 – Prototipo del caso de uso cerrar sesión.



Anexo B.3 – Prototipo del caso de uso Cambiar Contraseña.

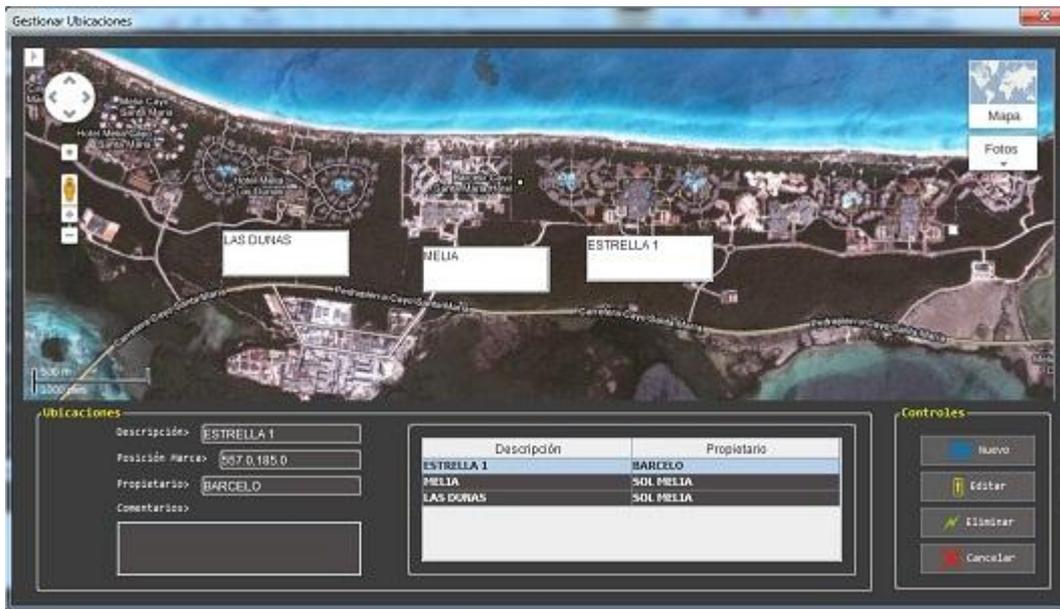
The screenshot shows a dialog box titled 'Autenticar Usuario' with a close button (X). Inside the dialog, the title is 'Cambiar Contraseña'. The form is organized into sections:

- Usuario**: A label 'Usuario>' followed by a text input field containing 'Admin'.
- Contraseña**: A label 'Contraseña>' followed by a password input field with six dots.
- Confirmar**: A label 'Confirmar>' followed by a password input field with six dots.
- Controles**: Two buttons at the bottom: 'Aceptar' (with a lightning bolt icon) and 'Cancelar' (with a red X icon).

Anexo B.4 – Prototipo del caso de uso visualizar ubicaciones.



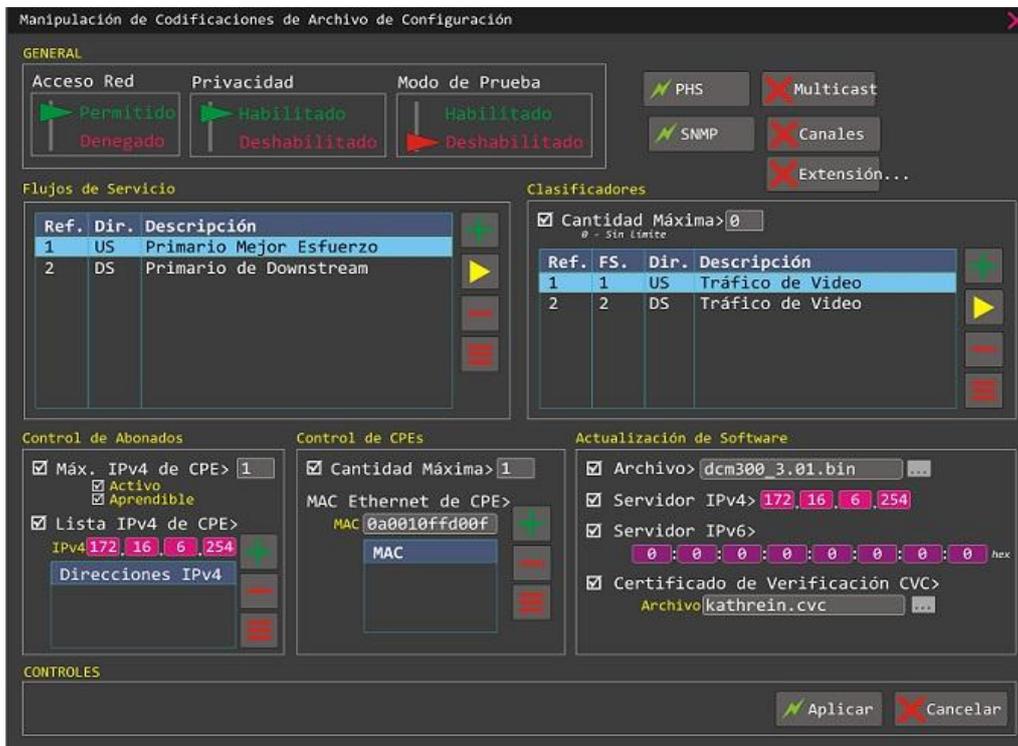
Anexo B.5 – Prototipo del caso de uso Gestionar archivo de configuración.



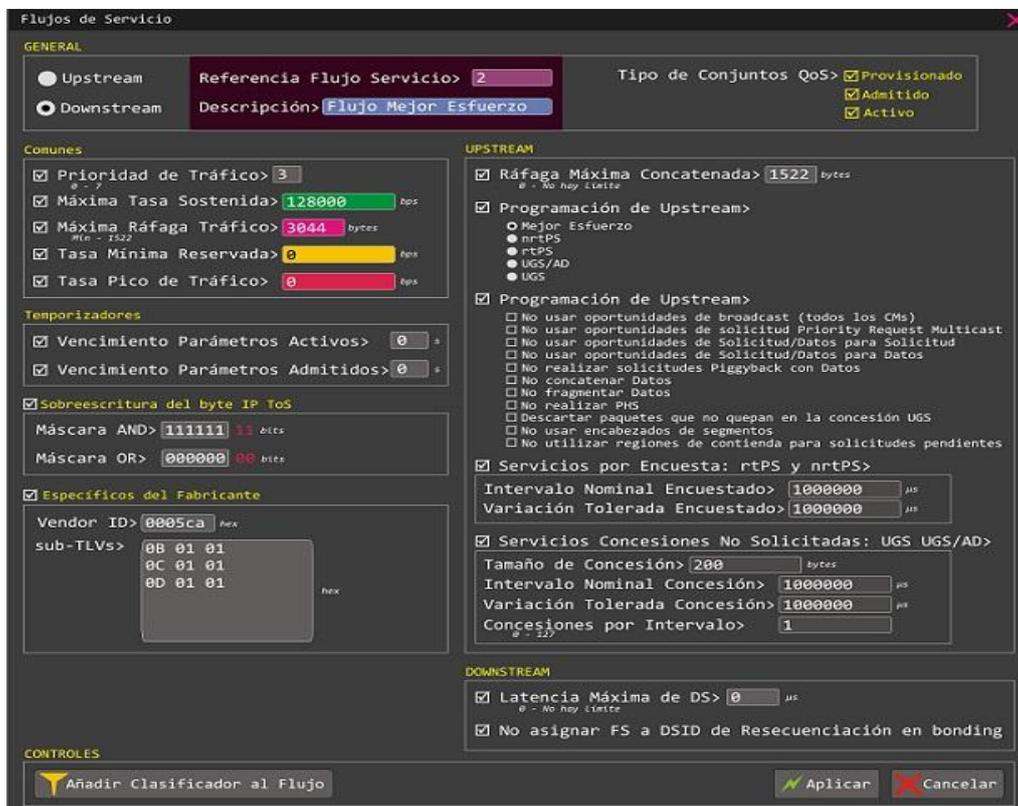
Anexo B.6 – Prototipo del caso de uso Gestionar archivo de configuración.



Anexo B.7 – Prototipo del caso de uso Gestionar archivo de configuración.



Anexo B.8 – Prototipo del caso de uso Gestionar archivo de configuración.



Anexo B.9 – Prototipo del caso de uso Gestionar archivo de configuración.

Clasificadores

GENERAL

Upstream Referencia de Clasificador> 2 Prioridad> 3 Activado
 Downstream Referencia Flujo Servicio> 2 Desactivado

IPv4 IPv6

ToS> Inferior 00 hex
 Superior 3F hex
 Máscara 00 hex

Clase de Tráfico> Inferior 00 hex
 Superior 3F hex
 Máscara 00 hex

Protocolo IP> 3

Etiqueta de Flujo> 00000000 hex

Fuente> 10.0.6.0 / 255.255.255.255 Fuente> 0:0:0:0:0:0:0:0 hex
 1111:1111:1111:1111:1111:1111:1111:1111
 Longitud del Prefijo 128
 n bits = 1

Destino> 10.0.6.0 / 255.255.255.255 Destino> 0:0:0:0:0:0:0:0 hex
 1111:1111:1111:1111:1111:1111:1111:1111
 Longitud del Prefijo 128
 n bits = 1

Puerto Fuente> Inicial 0 Final 65535
 Puerto Destino> Inicial 0 Final 65535

Ethernet LLC IEEE 802.1 P/Q

MAC Destino> Destino 0005cad00faa hex
 Máscara ffffffff hex

Prioridad 802.1P> Inferior 2
 Superior 7

MAC Fuente> Fuente 0005cad00faa hex
 Máscara ffffffff hex

VLAN_ID 802.1Q> 251

EtherType/DSAP/MacType> Máscara de Interfaces de CM - CMIM

Tipo 2-Ethertype eprot1 eprot2
 00 A2 hex

2da CM Reservados
 P110 1P 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 bits

CONTROLES

Aplicar Cancelar

Anexo B.10 – Prototipo del caso de uso Gestionar archivo de configuración

Supresión Encabezados de Paquetes - PHS

GENERAL

Referencia de Clasificador> 2
 Referencia Flujo Servicio> 2

ESPECÍFICO

Valor del Encabezado a Suprimir> En el 05 comienza en el 1er byte de La PDU
En el 05 comienza en el byte 13 de La PDU.
 P10F 0c 21 0d ff 00 00 00 dd a8 c4 0f hex

Máscara> 1 - Suprimir byte
 P10M 1 1 1 1 0 0 0 1 1 1 1 bits

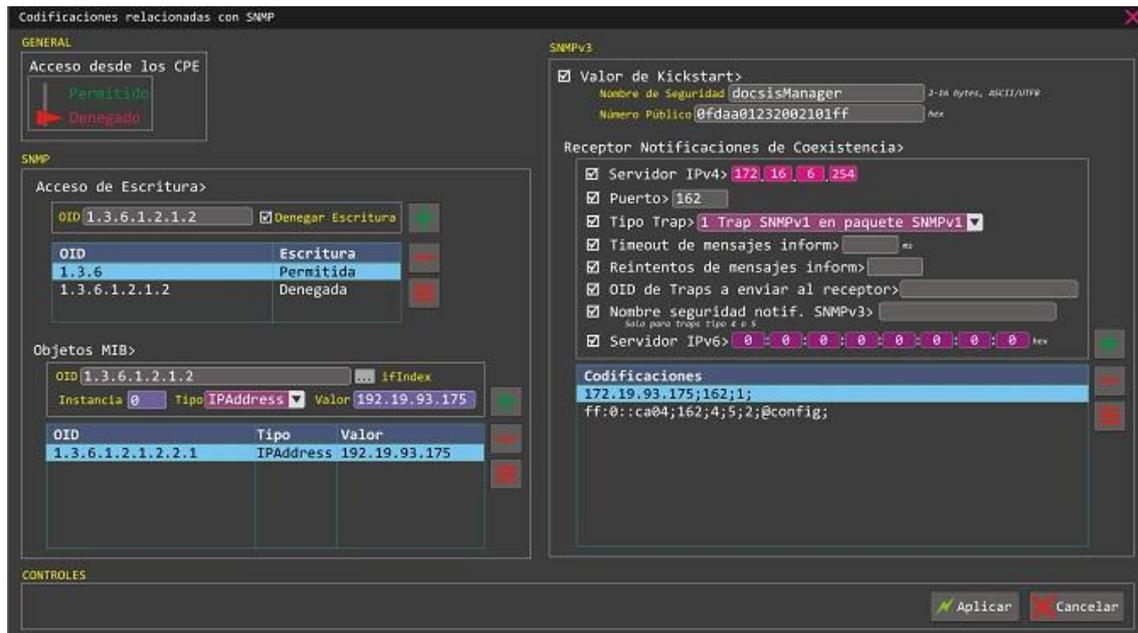
Tamaño> ANSI 11

Verificación de Encabezado contra PHSF> ANSI Desactivar

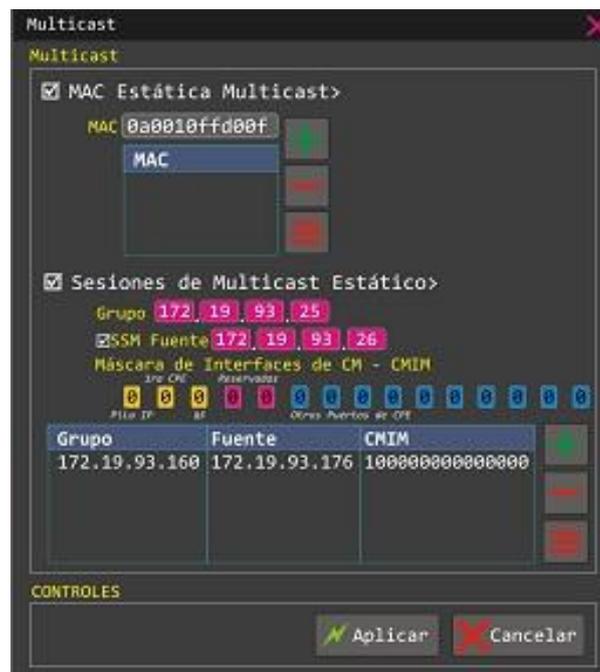
CONTROLES

Aplicar Cancelar

Anexo B.11 – Prototipo del caso de uso Gestionar archivo de configuración



Anexo B.12 – Prototipo del caso de uso Gestionar archivo de configuración



Anexo B.13 – Prototipo del caso de uso Gestionar archivo de configuración

Configuración de Canales

Canales

Identificador de Canal de Upstream> 3

Lista de Canales de Downstream>

Canal Simple>

Frecuencia	Timeout
Frec. 140000000 Hz <small>Múltiplo de 62.500</small>	Timeout 0 s <small>0 - Sin Límite</small>

Rango de Canales>

Frec. Inicial	Frec. Final	Timeout	Paso
140000000 Hz <small>Múltiplo de 62.500</small>	210000000 Hz <small>Múltiplo de 62.500</small>	0 s <small>0 - Sin límite</small>	6 MHz

Inicio	Fin	Paso	Timeout
--------	-----	------	---------

Búsqueda por defecto>

Timeout 0 s
0 - Sin Límite

Asignación de Canales

Tipo	USID
TX	USID 1

Tipo	USID o Frecuencia	DS
------	-------------------	----

CONTROLES

Aplicar Cancelar

Anexo B.14 – Prototipo del caso de uso Gestionar archivo de configuración

Campo de extensión DOCSIS

Información General de Extensión

Identificador de Tipo de Servicio> basic ASCII 1-16

ID Política Balance Carga> 1

Prioridad Balance Carga> 1

ID Grupo Balance Carga> 1

Extensión Clase Alineación> 1000111100001111 HEX

Verificación de Direcciones Fuente (SAV)

Nombre Grupo SAV en el CMTS> grupo1

Reglas de Prefijo>

IPv4> 172.16.6.254

IPv6> 0:0:0:0:0:0:0:0 hex

Longitud Prefijo 00 hex

Dirección IPv4 o IPv6	Long. Prefijo
-----------------------	---------------

Autorización de Unión a Multicast IP

Nombre Perfil en el CMTS>

Nombre perfil ASCII 1-15

Perfiles

Reglas de Sesiones Estáticas>

Prioridad 1 Acción Permitir

FUENTE

IPv4> 172.16.6.254

IPv6> 0:0:0:0:0:0:0:0 hex

Longitud Prefijo 00 hex

GRUPO

IPv4> 172.16.6.254

IPv6> 0:0:0:0:0:0:0:0 hex

Longitud Prefijo 00 hex

P	A Fuente IPv4 o IPv6	Pref. Fuente IPv4 o IPv6	Pref.
---	----------------------	--------------------------	-------

Cantidad Máxima de Sesiones Multicast> 1

CMTS MIC Extendido

Tipo HMAC> MD5 HMAC

Mapa de Bits>

TLV Primario 55

TLV Incluidos en el Cálculo

Digest Explicito>

0f ff a0 b9 01 00 35 42 41 96 dd c4

Específicos del Fabricante

Vendor ID> 0005ca hex

Tipo 14 Valor 14 hex

T	L	V
---	---	---

L2VPN y VLAN

Vendor ID> ffffff hex

Identificador VPN 0234560001 hex

Encapsulación MSI

Etiqueta IEEE 802.1Q 012C hex

CONTROLES

Aplicar Cancelar

Anexo B.15 – Prototipo del caso de uso Gestionar usuarios.

Gestionar Usuarios

Gestionar Usuario

Lista Usuarios

Usuario	Privilegio	Nombre	Información Adicional
admin	Administrador	Deivis Gonzalez Gonzalez	Administrador del Sistema

Usuarios

Nombre> Usuario>

Información Adicional> Contraseña>

Privilegios> Confirmar>

Controles

Anexo B.16 – Prototipo del caso de uso Gestionar inventario de equipos DOCSIS.

Gestionar Inventario

Todos

MAC	No. Inv.	Ubicación	Tipo	Marca	Modelo	No. Serie
0005cadf2c01	216345	LAS DUNAS - 3401	CM D3.0	Kathrein	DCM300	01221A5
0005cadf2c44	216346	LAS DUNAS - 3402	CM D3.0	Kathrein	DCM300	01455A5
0005cadf8988	216347	LAS DUNAS - 3403	CM D3.0	Kathrein	DCM300	01890A5
0005ca8d84ec	216347	LAS DUNAS - 3404	CM D3.0	Kathrein	DCM300	01890A5

Detalles

Fabricante

MAC>

Marca>

Modelo>

No Serie>

Tipo>

No Inventario>

Operador

Ubicación

ubicación> -

Controles

Agregar Manteniendo datos exceptos No Duplicados

Anexo B.17 – Prototipo del caso de uso Gestionar fichero DHCP.

Generalidades

Subred CM> 10 - 0 - 6 - 0 / 255 - 255 - 255 - 0

Servidor TFTP> 172 - 19 - 93 - 175

Servidor T00> 172 - 19 - 93 - 175

Servidor Logs> 172 - 19 - 93 - 175

Puerta Enlace> 10 - 0 - 6 - 1

Lease

Default> 7200 Min> 3600 Max> 86400

Corrimiento Tiempo UTC (Segundos)> -18000

Cantidad de Direcciones Disponibles> 658

Subred Denegada> 172 - 19 - 93 - 160 / 255 - 255 - 255 - 224

Subred CPE> 172 - 16 - 6 - 0 / 255 - 255 - 255 - 0

Servidor DNS> 172 - 19 - 93 - 175 Dominio> telecable

Servidor NTP> 172 - 19 - 93 - 175

Servidor Logs> 172 - 19 - 93 - 175

Puerta Enlace> 172 - 16 - 6 - 1

Lease

Default> 7200 Min> 3600 Max> 86400

Corrimiento Tiempo UTC (Segundos)> -18000

Rango Direcciones> 172 - 16 - 6 - 2 / 255 - 255 - 255 - 0

Cantidad de Direcciones Disponibles> 658

Atencion: Este debe ser la direccion ip del DOCSIS MAC del CMTS para los CMTS

Atencion: Este debe ser la direccion ip del DOCSIS MAC del CMTS para los CMTS

Controles de Perfil

Fichero1 Fichero> Fichero1 Guardar Cargar

Controles de Servicio

Reiniciar Iniciar Detener Cancelar

Anexo B.18 – Prototipo del caso de uso Gestionar fichero DHCP.

Asignacion de Direcciones de IP

Servicio DHCP - Asignacion de Direcciones de IP

Todos

MAC	Inventario	Ubicacion
0005cdf2c01	216345	LAS DUNAS - 3401
0005cdf2c44	216346	LAS DUNAS - 3402
0005cdf8988	216347	LAS DUNAS - 3403

Asignacion en Grupo

Nombre Fichero> Fichero1

IP Inicial> 10 - 14 - 16 - 2

IP Final> 10.14.16.2

Archivo> CM.cfg

Asignados

MAC	IP	Archivo	Inventario	Ubicacion
0005ca8d84ec	10.0.6.200	CM.cfg	216347	LAS DUNAS - 3404

Controles de Perfil

Guardar Cancelar

Anexo F.1 Prueba de Kolmogorov-Smirnov para una muestra.

Prueba de Kolmogorov-Smirnov para una muestra			TiempoAntes	Tiempo Despues
N			15	15
Parámetros normales ^{a,b}	Media		18,2000	6,4000
	Desviación típica		4,63219	2,87352
Diferencias más extremas	Absoluta		,222	,245
	Positiva		,222	,154
	Negativa		-,178	-,245
Z de Kolmogorov-Smirnov			,859	,947
Sig. asintót. (bilateral)			,452	,331

a. La distribución de contraste es la Normal.

b. Se han calculado a partir de los datos.

Anexo F.2 Muestras relacionadas.

Estadísticos de muestras relacionadas

	Media	N	Desviación típ.	Error típ. de la media
Par 1 TiempoAntes	18,2000	15	4,63219	1,19603
TiempoDespues	6,4000	15	2,87352	,74194

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	11,80000	4,07431	1,05198	9,54372	14,05628	11,217	14	,000