



UNIVERSIDAD POLITÉCNICA DE CATALUNYA

FACULTAD DE INGENIERÍA

ESCUELA INGENIERÍA DE TELECOMUNICACIONES



**DISEÑO E IMPLEMENTACIÓN DE UNA HERRAMIENTA DE
VALIDACIÓN E INVENTARIADO DE ROUTERS**

DESIGN AND IMPLEMENTATION OF A VALIDATION AND ROUTERS
INVENTORYING TOOL

PROYECTO FINAL DE CARRERA

Presentado ante la

UNIVERSIDAD POLITÉCNICA DE CATALUNYA

como parte de los requisitos para optar al título

INGENIERO EN TELECOMUNICACIONES

REALIZADO POR: Andrés Alberto Quintero Valero

DIRECTOR: Jesús Murcia

CO-DIRECTOR: Beatriz Otero

FECHA: Barcelona, Octubre2016



UNIVERSIDAD POLITÉCNICA DE CATALUNYA
FACULTAD DE INGENIERÍA
ESCUELA INGENIERÍA DE TELECOMUNICACIONES



**DISEÑO E IMPLEMENTACIÓN DE UNA HERRAMIENTA DE
VALIDACIÓN E INVENTARIADO DE ROUTERS**

**DESIGN AND IMPLEMENTATION OF A VALIDATION AND ROUTERS
INVENTORYING TOOL**

REALIZADO POR: Andrés Alberto Quintero Valero

DIRECTOR: Jesús Murcia

CO-DIRECTOR: Beatriz Otero

FECHA: Barcelona, Octubre 2016



Resumen del proyecto

El presente trabajo trata sobre el desarrollo de una herramienta de soporte automático para la validación de equipos instalados, explicando cada una de las evaluaciones y procesos que deben realizarse, así como la metodología elegida y su funcionamiento general. Todo esto permitirá aumentar la eficiencia, disminuir el trabajo sobre los recursos del proyecto y determinar las siguientes acciones a tomar para cada equipo de la manera más fácil posible.

Además también se incluye un proceso de inventariado automático de cada *router*, con el objetivo de mantener un control histórico sobre aquellos que han sido instalados cada día, facilitando la obtención de resúmenes y análisis de datos . El registro de un equipo en el inventario debe realizarse siguiendo un formato específico de columnas y datos.

La implementación de esta herramienta constituye una considerable reducción de recursos para este proceso. Anteriormente se requerían de dos personas del equipo de provisión del servicio dedicando varias horas a las pruebas y evaluaciones sobre cada equipo instalado, verificando cuáles están listos para facturar y cuáles acciones tomar sobre aquellos con alguna tarea pendiente o mal configurados. También manualmente se recolectaban datos de diferentes sistemas para completar el formato requerido para inventariar cada equipo. Gracias a esta automatización se pueden utilizar este tiempo y recursos en otras actividades aumentando la eficiencia del proyecto.

El trabajo presenta una serie de análisis de los resultados obtenidos con propuestas a futuro para realizar una optimización de la herramienta con funciones más amplias y de soporte hacia otros equipos del proyecto.

Palabras clave: Validación, Inventariado, *Router*, Provisión del servicio.



Abstract

This paper is about the development and implementation of an automated tool for the validation of installed routers, explaining each of the evaluations and processes to be performed, as well as the chosen methodology and overall function. This will increase the efficiency, reduce work on others resources and determine the following actions to take for each equipment in the easiest possible way.

Furthermore, a process of automatic inventorying for each router is also included, with the objective of maintaining a historical control over those routers that have been installed every day, facilitating summaries and analysis of data. The registration of a router in the inventory should be done following a specific format of data columns.

The implementation of this tool presents a considerable reduction in resources for this process. Previously they required two person of the service provision team dedicating many hours in manually performing tests and evaluations on each installed router, checking which are ready to bill and what actions to take on those with a pending task or bad configurations. Also, they had to manually make the collection of data from different systems to complete the required format for the inventory of each equipment. Thanks to this tool, this time and resources can be used on other activities, increasing the efficiency of the project.

The paper presents an analysis of the obtained results with future proposals for optimization of the tool with more functions and support to other project teams.

Keywords: Validation, Inventory, Router, Service provision.

Resum del projecte

El present treball tracta sobre el desenvolupament d'una eina de suport automàtic per a la validació de equips instal·lats, explicant cada una de les avaluacions i processos que han de fer-se, així com la metodologia escollida i el seu funcionament general. Tot això permetrà augmentar la eficiència, disminuir el treball sobre els recursos del projecte i determinar les següents accions a prendre per a cada equip de la manera més fàcil possible.

A més també s'ha d'incloure un procés de inventariat automàtic per a cada router, amb el objectiu de mantindre un control històric sobre aquells que han sigut instal·lats cada dia, facilitant la obtenció de resums i anàlisis de dades.

La implementació d'aquesta eina constitueix una considerable reducció de recursos per a aquest procés. Anteriorment es requeria de dues persones en l'equip de proveïment del servei dedicant-hi varies hores a les probes i avaluacions sobre cada equip instal·lat, verificat quins estan llestos per factura i quines accions prendre sobre aquells amb alguna tasca pendent o mal configurats. També manualment recol·lectaven dades de diferents sistemes per completar el format requerit per inventariar cada equip. Graciés a aquesta automatització es poden utilitzar aquest temps i recursos en altres activitats augmentar la eficiència del projecte.

El treball presenta una sèrie de anàlisis dels resultats obtinguts amb propostes a futur per realitzar una optimització de la eina amb funcions més amples i de suport a altres equips del projecte.

Paraules clau: validiació, inventari, router, proveïment del servei.



Con la aprobación de la tutora del proyecto Beatriz Otero, este proyecto fue realizado siguiendo los requisitos de estructura de presentación del Proyecto Final de Carrera que exige la Universidad Católica Andrés Bello (Venezuela), por encontrarme en modalidad de estudiante de intercambio, y con la finalidad de cumplir con los requerimientos en ambas instituciones para optar al título de Ingeniero Superior en Telecomunicaciones.



Dedicatoria

A mi madre, por estar incondicionalmente en las buenas y malas, brindándome el apoyo y amor necesarios, así como un buen consejo incluso cuando no lo pedía.

A mis hermanos, por sentirse siempre tan orgullosos de mí como yo de ellos.

A Katherine Palazzese, por haberme ayudado a no rendirme y regañarme cuando era necesario. Enseñándome que sin sacrificios las cosas no llegan solas.

A mi padre, quién me daba ánimos e insistía en completar este trabajo, esto es para ti. Te Amo y te extrañaré siempre.

Andrés Quintero.



Agradecimientos

Sinceramente, son muchas las personas a las cuales tengo que agradecer, debido a que sin su apoyo no hubiera logrado esta meta, y todos los objetivos propuestos para la ejecución de este trabajo. Primeramente agradecer a Dios, por guiarme siempre por el camino correcto y me dio la fuerza para culminar esta etapa, confiando en su palabra.

A mi familia, que me demostraron que la distancia es solo una medida de separación cuando realmente se quiere estar presente. Gracias por el apoyo en cada derrota y dificultad y por brindarme la oportunidad de vivir en una ciudad como Barcelona. Por todo el ánimo, amor y esperanzas que siempre me dan.

A Katherine Palazzese, por indicarme que no debo rendirme ni abandonar algo que tanto esfuerzo me ha llevado, empujándome a dar ese último paso ofreciéndome siempre su apoyo y ayuda.

A la Universidad Católica Andrés Bello, porque gracias a su formación durante 4 años me enseñó mucho más que conocimientos adquiridos en un salón de clases. Fueron esos conocimientos y valores transmitidos que me permitieron afrontar este reto.

A mi tutora Beatriz Otero, por apoyarme en la realización de este trabajo, a pesar de los problemas y complicaciones que implicaba, por brindarme confianza y ayuda para poder culminar esta etapa y obtener mi título.

Por último, a la Universidad Politécnica de Catalunya, por brindarme la oportunidad de realizar este convenio, que más allá de una gran formación profesional, me permitió tener contacto con otras culturas que enriquecieron mi vida personal.



Índice General

Resumen del proyecto	3
Abstract	4
Resum del projecte	5
Dedicatoria	7
Agradecimientos.....	8
Índice General	9
Índice de Tablas	15
Índice de Figuras	16
Índice de Gráficos	17
Introducción	18
CAPÍTULO I.....	20
Planteamiento del Proyecto	20
I.1.- Planteamiento del Problema	20
I.2.- Objetivos.....	21
I.2.1 Objetivo General	21
I.2.2.- Objetivos Específicos	21
I.3.- Alcance	22
I.4.- Limitaciones	22
I.5.- Justificación y Motivación.....	23
I.6 Requisitos funcionales	24
CAPÍTULO II	26



Marco teórico	26
II.1.- Administración del proyecto	26
II.1.1 Everis	26
II.1.2 Orange España	27
II.1.3 Business Process Outsourcing (BPO).....	27
II.1.3 BPO informático en España	28
II.1.4 Importancia de <i>Business Process Outsourcing</i> para las empresas	30
II.1.5 Everis BPO	32
II.1.6 Provisión del servicio	33
II.2 Sistemas y Bases de Datos.....	35
II.2.1.- Lenguajes	35
II.2.1.1.- Elementos de un lenguaje.....	36
II.2.1.2.- El Lenguaje SQL y el Modelo Relacional	36
II.2.2 SQL Server:	41
II.2.3 Lotus Notes:.....	42
II.2.4 WOM:.....	43
II.2.5 Access:.....	44
II.2.6 Gestión de Circuitos (GESCIR):	44
II.2.7 HP <i>Inventory</i> :.....	44
II.2.8 NA <i>Inventory</i> :.....	46
II.2.9 Tacacs	46
II.2.10 Confluence.....	47
II.3 Términos relacionados al Proyecto.....	47
II.3.1 Router	47



II.3.2 Velocidad de sincronismo	48
II.3.3 DomainNameService DNS	48
II.3.4 Ancho de Banda.....	49
II.3.5 Servidor	49
II.3.6 Interfaces	50
II.3.7 Equipo Local del Cliente(CPE)	51
II.3.8 Página Web.....	52
II.4 Lenguajes de Programación.....	53
II.4.1 Vb.net	53
II.4.2 SSH.....	54
II.4.3 ASP.....	55
II.4.4 Javascript	56
II.4.5 HTML.....	57
II.4.6 CSS	57
CAPÍTULO III	59
Metodología	59
Fase I: Proceso de investigación y preparación.	59
Fase II: Página de control, gestión y soporte de PDS dentro de la Web	59
Fase III: Modelado de Datos de entrada al proyecto.....	60
Fase IV: Procesos de carga automáticos de los Inventarios.....	60
Fase V: Desarrollo del validador e inventariado	60
Fase VI: Pruebas.....	61
Fase VII: Elaboración del PFC.....	61
CAPÍTULO IV	62



Desarrollo	62
IV.1.-Fase I: Proceso de investigación y preparación.....	62
IV.1.1. Entrega de requerimientos	62
IV.1.2. Lógica de los procesos de validaciones	65
IV.1.3. Lenguajes de programación y herramientas a utilizar	67
IV.1.4 Valoración temporal.....	68
IV.2.-Fase II: Página de control, gestión y soporte de PDS dentro de la Web	69
IV.2.1 Estructura de la página.....	69
IV.2.1.1 Información WOM.....	69
IV.2.1.2 Estado.....	70
IV.2.1.3 Instalación- Datos	70
IV.2.1.4 Instalación- Accesos y CPes.....	72
IV.2.1.5 Instalación- RFS.....	74
IV.2.1.6 Comentarios y Archivos.....	75
IV.2.1.7 Gestión	75
IV.2.2 Cambios a nivel de código	76
IV.3.-Fase III: Modelado de Datos de entrada al proyecto	77
IV.3.1. Tabla del validador.....	77
IV.3.2. Tabla de inventario.....	78
IV.3.3. Procedimiento de exportación de datos entre las BBDD.....	79
IV.3.3.1 Carga_PDS_Web_SQL.....	79
IV.3.3.2 Insert_Into_IS2.....	79
IV.3.4. Modificación tabla y proceso de carga WOM	80
IV.3.5. Modificación tabla y proceso de carga GESCIR	80



IV.4.-Fase IV: Procesos de carga automáticos de los Inventarios	81
IV.4.1. Tabla y proceso de carga NA	81
IV.4.2. Tabla y proceso de carga HP.....	83
IV.5.- Fase V: Desarrollo del validador e inventariado	83
IV.5.1 Tarea de ejecución y log	86
IV.5.2Ejecución procedimiento.....	87
IV.5.3Evaluaciones técnicas por comando SSH	87
IV.5.4 Vista SQL de cruce.....	89
IV.5.5 Inventariado	89
IV.5.6 Evaluaciones finales.....	90
IV.5.7 Actualización de campos	90
IV.6.- Fase VI: Pruebas	91
IV.6.1 Pruebas de la página de soporte datos PDS	91
IV.6.2 Pruebas del proceso de exportación	91
IV.6.3 Pruebas de los procesos de carga de los inventarios	92
IV.6.4 Pruebas del proceso de inventariado	92
IV.6.5 Pruebas del validador	92
IV.7.- Fase VII:Elaboración del PFC	93
CAPÍTULO V	94
Resultados obtenidos.....	94
CAPÍTULO VI.....	96
Conclusiones y Recomendaciones	96
VI.1.- Conclusiones.....	96
VII.2.- Recomendaciones.....	97



BIBLIOGRAFÍA.....	99
Anexos.....	100



Índice de Tablas

Tabla 1: Ventajas y desventajas del BPO para las empresas	31
Tabla 2: Valoración estimada de tiempo.....	68
Tabla 3: Diseño tabla visitas.....	72
Tabla 4: Diseño tabla de accesos	73
Tabla 5: Diseño tabla de CPE	73
Tabla 6: Diseño tabla de RFS	74

Índice de Figuras

Figura 1: Validaciones para la facturación de un equipo.....	20
Figura 2: El proceso de provisión de servicio a un cliente [6].....	34
Figura 3: Menú inicial del IBM Lotus Note.....	42
Figura 4: Bases de datos Lotus Notes del proyecto.	43
Figura 5: Base de datos GESCIR.....	44
Figura 6: Opciones asociadas a cada equipo en HP <i>Inventory</i>	45
Figura 7: Alta de un equipo en HP <i>Inventory</i>	45
Figura 8: Relación servidor y navegador web [10].....	53
Figura 9: Funcionamiento del protocolo SSH [11].....	54
Figura 10: Ventajas del uso de CSS en el desarrollo de páginas web [10].....	58
Figura 11: Proceso general del proyecto.....	63
Figura 12: Método de las evaluaciones del validador.....	67
Figura 13: Menú inicial Web Provisión.....	69
Figura 14: Datos relacionados al Proyecto.....	70
Figura 15: Datos de la instalación.....	71
Figura 16: Datos del equipo y sus accesos.....	72
Figura 17: Datos de la RFS.....	74
Figura 18: Comentarios y Archivos.....	75
Figura 19: Tabla de Gestión.....	76
Figura 20: Web de NA.....	81
Figura 21: Selección <i>Devices</i>	82
Figura 22: Selección <i>Inventory</i>	82
Figura 23: Procesos de los datos de una instalación.....	84
Figura 24: Partes del programa de validación e inventariado.....	86



Índice de Gráficos

Gráfico 1: Principales áreas en las que se implementa el <i>Outsourcing</i> [3].....	28
Gráfico 2: Aumento del uso del <i>Outsourcing</i> según la VI encuesta de Adecco <i>Outsourcing</i> 2015 [3].....	29
Gráfico 3: Perfiles de la empresas que utilizan <i>Outsourcing</i> informático según la VI encuesta de Adecco <i>Outsourcing</i> 2015 [3].	30
Gráfico 4: Ventajas del uso de la Externalización en las empresas [2].	32
Gráfico 5: Evolución de ventas de Everis BPO [5]	33



Introducción

Con los años, las tecnologías de la información y comunicación (TIC) han permitido que una gran parte de las empresas pueda distribuir sus funciones y trabajo en diferentes lugares, y especialmente, que sea hecho por otras empresas especializadas con experiencia, que aporten una mejora en el desarrollo de sus funciones, disminuyendo sus costos y gastos, aumentando su eficiencia.

Las TIC son un instrumento de gran importancia para la organización interna de cualquier proyecto o negocio. Con una utilización correcta de las TIC se puede obtener un importante ahorro de tiempo y recursos al simplificar, automatizar y agilizar los procesos de gestión, toma de decisiones y tareas rutinarias, dejando una mayor cantidad de tiempo y recursos para invertir en actividades productivas. De igual manera se puede mejorar el contacto directo con la clientela y optimizar la gestión del negocio mediante aplicaciones y herramientas informáticas que permitan un mayor control sobre aquellas variables, tareas y procesos que intervienen en el negocio.

En la actualidad, muchas de las tareas y funciones comunes de las empresas suelen externalizarse, y una de las áreas que suelen ser más susceptibles de tercerización son la de administración y gestión de la tecnología, así como la provisión del servicio, la contabilidad y las finanzas. Dicha subcontratación, o externalización, de funciones de los procesos de negocio en proveedores de servicio es lo que se conoce como, en inglés, *Business Process Outsourcing* (BPO).

Everis es una consultora que tiene proyectos de negocio, tecnología y externalización en diferentes sectores. Entre los proyectos de BPO en el sector de las telecomunicaciones se encuentra uno con Orange, empresa francesa de telecomunicaciones muy conocida en el ámbito europeo. La provisión del servicio es



una de las áreas de mayor complejidad e importancia en el proyecto, en la que actúan diferentes procesos y gestiones que pueden variar dependiendo del tipo de ítem que requiera un cliente específico, y otras variables que pueden aumentar la dificultad de la provisión. Una de las etapas finales de la provisión del servicio es la validación de la correcta instalación y gestión de los equipos instalados. Durante esta etapa se hacen diferentes evaluaciones para determinar qué tareas y qué funciones no se han completado para cada instalación de manera que finalicen lo antes posible de forma correcta para poder ser facturados los equipos.

Durante este proceso se han estado ocupando muchos recursos y tiempo en la validación y comprobación de los equipos instalados. A fin de ahorrar estos gastos y poder utilizar ese tiempo y recursos en otras actividades se ha planteado desarrollar una herramienta automática que realice las tareas de inventariado y validación de los equipos instalados cada día.

El presente trabajo tiene como propósito la realización de este proyecto de validación e inventariado automática de equipos instalados. El trabajo consta de 6 capítulos ordenados como sigue: el capítulo 1 establece el problema a tratar, los objetivos a alcanzar del proyecto, su alcance y posibles limitaciones, así como la motivación para realizarlo; el capítulo 2 describe el marco teórico, donde se explican los conceptos y los conocimientos técnicos necesarios para el desarrollo del proyecto; el capítulo 3 presenta la metodología a desarrollar durante la investigación; el capítulo 4 detalla con precisión las funciones y las tareas realizadas en cada fase, establecidas previamente en la metodología; el capítulo 5 presenta los resultados obtenidos y su análisis. Finalmente, el capítulo 6 corresponde a las conclusiones y recomendaciones del proyecto.

CAPÍTULO I

Planteamiento del Proyecto

El presente capítulo establece las bases fundamentales del trabajo, especificando el problema a tratar y sus características, determinando los alcances y posibles limitaciones, dejando claro los objetivos y dirección del presente trabajo.

I.1.- Planteamiento del Problema

Para poder facturar un equipo instalado, se debe hacer previamente una serie de validaciones para verificar que la instalación y configuración del equipo se realizaron correctamente, así como comprobar que las tareas de gestión e inventariado estén finalizadas. Éstas validaciones son las siguientes:



Figura 1: Validaciones para la facturación de un equipo.



En un proyecto en el que suelen instalarse cientos de *routers* mensuales se requiere de una gran cantidad de tiempo y de recursos para validar que dichos equipos funcionen correctamente, comprobando sus estados diariamente para poder tomar acciones correspondientes a cada equipo con el fin de que sean facturables lo antes posible.

Por el motivo mencionado, se propone el desarrollo de una herramienta automática de validación diaria de los equipos instalados que aún no se hayan facturado para poder tener un control sobre los estados de cada *router*, saber cuáles se pueden facturar y qué acciones tomar con respecto a los que no estén aun finalizados, ahorrando recursos y aumentando la eficiencia del proyecto.

I.2.- Objetivos

I.2.1 Objetivo General

Diseñar e implementar una herramienta automática de validación e inventariado de equipos instalados.

Para alcanzar este objetivo será necesario realizar los siguientes objetivos específicos:

I.2.2.- Objetivos Específicos

- Implementar una herramienta que compruebe la conectividad, configuración DNS, usuario de TACACS, ancho de banda y velocidades de sincronismo para cada *router*.
- Automatizar la carga de los inventarios de HP y NA a la Base de Datos para poder comprobar diariamente equipos que hayan sido inventariados.
- Agregar a la Web de provisión del servicio (PDS) una herramienta de control de los resultados del validador de cada equipo.



- Automatizar un proceso que verifique tareas de gestión pendientes en el sistema WOM para saber qué equipos tienen tareas previas a la facturación sin finalizar.
- Automatizar el inventariado de los equipos instalados en el inventario de *routers* del proyecto.

I.3.- Alcance

En el presente trabajo se implementará una herramienta que permita evaluar los estados de las instalaciones de *routers* realizadas, incluyendo pruebas específicas para determinar qué acciones deben realizarse y corregirse. Sólo se darán resultados en forma de OK/KO para cada evaluación, no se harán correcciones automáticas, al menos en esta primera fase de desarrollo. Sólo se planteará el uso de la herramienta como un control de los estados para una mejor administración del tiempo y de los recursos. Se mostrará por medio de la Web los resultados de la validación de cada *router*. Se realizarán los procesos de cargas automáticas de datos que sean necesarios para la evaluación de los equipos. La validación automática se realizará una vez por día, por medio de una tarea programada desde el servidor.

I.4.- Limitaciones

Por temas de privacidad de las empresas relacionadas con el proyecto, los datos que se mostrarán durante el presente trabajo son ficticios, específicamente datos relacionados con las instalaciones, clientes, equipos, bases de datos, nombres de tablas, etc.

Debido a que la frecuencia de la validación será diariamente, realizando las diferentes validaciones para los equipos instalados que aún no estén gestionados completamente, se pueden presentar KOs falsos, particularmente en las pruebas relacionadas a la conectividad del equipo, configuración DNS del *router*, autenticación en TACACS, velocidades de sincronismo y anchos de banda, es decir, aquellas



evaluaciones relacionadas a la parte técnica de la configuración del equipo durante la instalación, dado que si por alguna razón algún se encuentra apagado, reiniciando, o sin conectividad justamente durante el momento que el automatismo vaya a evaluarlo, dará KO en todas estas pruebas ya que éstas se realizan por medio de comandos al equipo por código SSH.

Otra limitación es que por medio de la web de HP *Inventory*, no se puede obtener un inventario histórico de equipos dado de alta, a diferencia de la de NA. Por lo que esta parte del proceso no será 100% automática, sino que el proceso de carga diario del inventario de HP *Inventory* a la BBDD dependerá de una extracción enviada a un buzón de correo del proyecto que contenga el inventario, y el proceso automático de carga lo importe en la correspondiente tabla de la BBDD.

I.5.- Justificación y Motivación

Además de aportar una herramienta de gran utilidad al proyecto que permita aumentar la eficiencia y facilitar la gestión de los equipos instalados, así como la facturación, también se justifica el desarrollo de esta herramienta con el fin de quitar carga de trabajo sobre el personal del equipo de PDS, así como mantener un mejor control sobre los estados de los ítems del proyecto.

A nivel de motivación personal, me ha sido de mucho interés este proyecto porque, si bien se trata de un desarrollo informático con mucha programación, está orientado sobre temas ligados a las telecomunicaciones, aplicando también comandos de configuración y acceso de *routers*. También es una motivación personal la gran cantidad de nuevo aprendizaje que implica este desarrollo, no sólo un nuevo lenguaje de programación sino una mayor comprensión sobre ley de negocio del proyecto y una mayor visión de los procesos de la parte de la provisión del servicio.



I.6 Requisitos funcionales

A continuación se presentan los requisitos funcionales que determinan lo que la herramienta debe proporcionar, es decir, lo que debe hacer el validador y cómo debe funcionar. Estos requisitos se sacan a partir de la toma de requerimientos que se describirá en la primera fase del desarrollo.

- La herramienta debe pasar por el validador diariamente los *routers* instalados durante el día cuyos datos se hayan cargado en la Web de *PdS*, así como aquellos equipos instalados en días anteriores pero que aún no se hayan facturados por no haber sido validados correctamente.
- La herramienta debe insertar un registro en el inventario de *routers* por cada acceso de cada equipo instalado, con el formato de datos debido.
- La herramienta debe mantener un control de cada paso por el validador de los *routers*, indicando para cada equipo: fecha de última validación, fecha de inventariado, número de iteraciones por el validador.
- El validador debe comprobar la conectividad de cada equipo a la red.
- El validador debe verificar la configuración DNS de cada *router*.
- El validador debe comprobar que cada equipo tenga autenticación en *Tacacs*.
- El validador debe verificar las velocidades de sincronismo y anchos de banda de cada *router*.
- El validador debe determinar para cada *router* si se ha dado de alta en las webs de los inventarios de HP y NA.
- El validador debe informar para cada *router* si en el sistema WOM hay tareas sin finalizar que impidan la facturación del equipo.
- La herramienta debe comprobar que para cada instalación se haya subido una plantilla de RFS a *Confluence*.
- La herramienta debe mostrar en la página Web de soporte de *PdS* los resultados del último paso por el validador para cada *router*.



- La herramienta se debe ejecutar automáticamente cada día desde el servidor, así como tener la posibilidad de ejecutarla manualmente en cualquier momento mediante un ejecutable *.bat* en el servidor.

CAPÍTULO II

Marco teórico

Una vez definido el enfoque y los objetivos del proyecto, se presentan a continuación diversos aspectos teóricos que le dan soporte. Todos estos conceptos mencionados anteriormente dan la capacidad de comprender todos los elementos que se analizarán más adelante en el desarrollo del proyecto.

II.1.- Administración del proyecto

II.1.1 Everis

Es la empresa de consultoría responsable del proyecto para el cual se desarrolla la herramienta sobre la que trata el presente trabajo. Es una empresa multinacional, de origen español, que abarca proyectos de desarrollo y estrategia de negocio, mantenimiento y desarrollo de aplicaciones tecnológicas y de *outsourcing*. Estos proyectos pueden ser de sectores variados, entre los cuales se mencionan las telecomunicaciones, banca, entidades financieras, industria, sanidad, administración pública, seguros, *utilities*, energía, entre otros.

Fundado en Madrid en 1996 como la filial en España de otra empresa consultora, *DMR Consulting*. En 2006 los directivos compran el 100% de capital de la compañía y le cambian el nombre a Everis. Actualmente opera en más de 13 países del mundo en continente Americano y Europeo. Desde 2013, fue adquirida por la empresa informática japonesa NTT Data, sin embargo, su nombre y muchas características de la empresa se mantienen.



II.1.2 Orange España

Legalmente llamada Orange *Espagne* S.A.U., es un operador de telefonía fija y móvil, internet y televisión por suscripción filial de la compañía multinacional de telecomunicaciones, de origen francés, Orange. Es el tercer operador en España.

Nace en 2006 tras la unión de todas las empresas españolas propiedad de France *Télécom* bajo la marca de Orange. Hasta ese entonces, el operador de telefonía fija e Internet era *Wanadoo* y el operador de telefonía móvil era Amena. Además de éstos, en 2007 Orange compró el operador fijo Ya.com y Deutsche Telekom.

II.1.3 Business Process Outsourcing (BPO)

La externalización de procesos de negocio (*Business Process Outsourcing* en inglés), considera la subcontratación de negocio a proveedores de servicios en base a un contrato en el que se garantiza la calidad de prestación por medio de la definición de indicadores de servicios. Se delega la gestión de los medios, en beneficio de la gestión de los niveles de servicio. Esta práctica conlleva un *partnership* estratégico entre cliente y proveedor, integrándose las actividades de BPO dentro del modelo de negocio del cliente. [1]

El BPO funciona sobre una base de continuidad en la operación de procesos que forman parte de la ruta clave de la cadena de entrega de valor, esto diferencia al BPO del *Outsourcing* tradicional.

Sin embargo, a pesar que la herramienta de *Outsourcing* puede abarcar la mayoría de procesos de la empresa, existen diversas áreas que por razones de seguridad y privacidad no se deben subcontratar, entre las cuales podemos nombrar: la administración de la planeación estratégica, la tesorería, el control de proveedores, la administración de calidad, el servicio al cliente y la distribución y ventas. [2]

Antagónicamente a lo anteriormente citado, las áreas más comúnmente contratadas para la externalización de procesos son las siguientes:

- **BPO de Administración y Finanzas:** incluye el análisis financiero, reporte y planeamiento financiero, contabilidad gerencial, gestión de tesorería y caja, pago y recibo de cuentas, administración de riesgo e impuestos.
- **BPO de Recursos Humanos:** especialmente centrado en el servicio de captación de candidatos, conocido como reclutamiento.
- **BPO de Servicios de *Customer Contact*:** *call centers*, servicio de atención al cliente, de reclamaciones, etc.
- **BPO de Servicios de Gestión de la Información:** control del flujo de la información, de los medios de soporte (hardware, software), así como todo lo relacionado con tecnologías de la información y la comunicación.

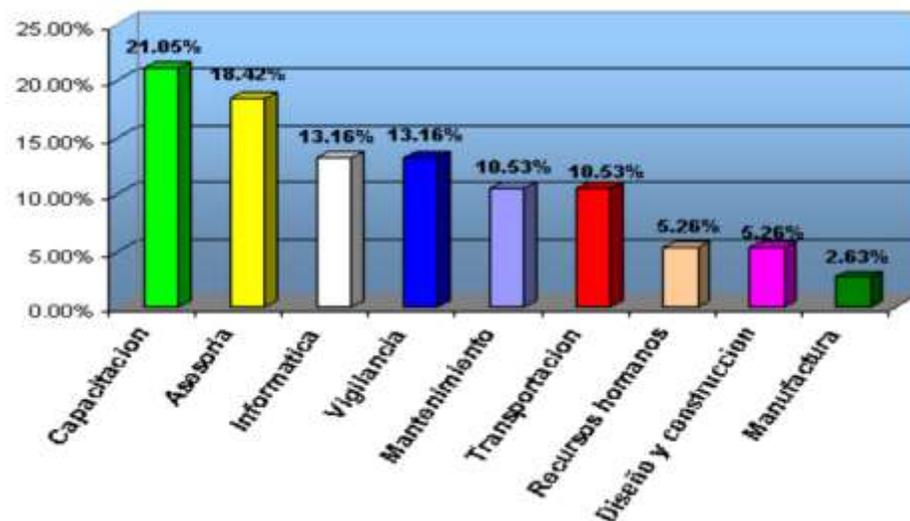


Gráfico 1: Principales áreas en las que se implementa el *Outsourcing* [3].

II.1.3 BPO informático en España

Según lo reflejado en 2015 por la VI encuesta de Adecco *Outsourcing*, el 64,6% de las empresas españolas han pensado en aumentar la externalización de servicios,

debido al ahorro en tiempo, costes e innovación tecnológica que esto supone. Todo esto apunta a que la externalización de servicios en las empresas va aumentando exponencialmente.

Las empresas energéticas son quienes creen que aumentarían el uso de la externalización de sus procesos. Seguidamente, muy cerca de estas, se encuentran las empresas del sector TIC, quienes apuestan por la expansión del *outsourcing* en el 71,7% de los casos como se puede ver en el gráfico a continuación:

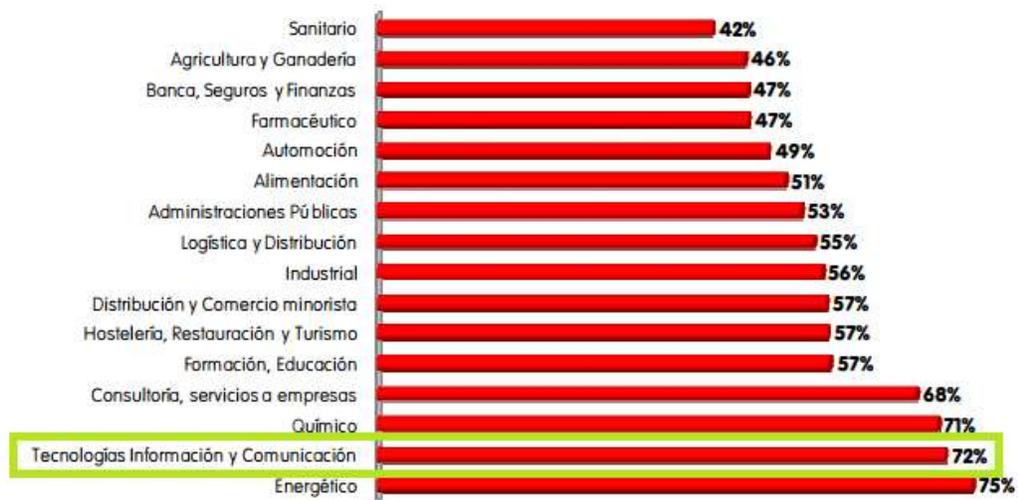


Gráfico 2: Aumento del uso del *Outsourcing* según la VI encuesta de Adecco *Outsourcing* 2015 [3].

Sin embargo, según este informe de Adecco, las organizaciones que utilizan el *outsourcing* informático a pesar de tratarse de grandes empresas (33.9%) no responden a ningún perfil en particular.

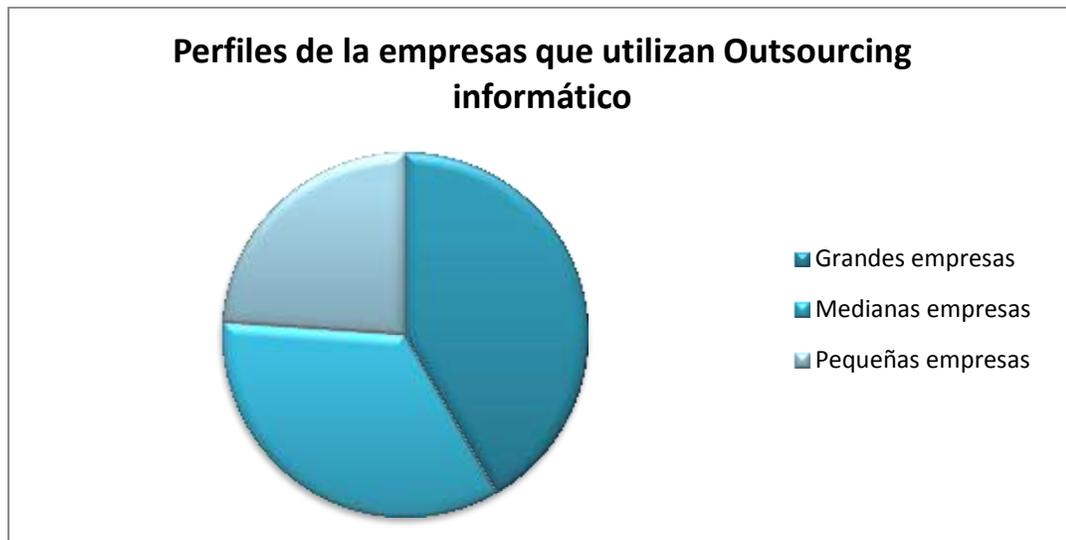


Gráfico 3: Perfiles de la empresas que utilizan *Outsourcing* informático según la VI encuesta de Adecco *Outsourcing* 2015 [3].

II.1.4 Importancia de *Business Process Outsourcing* para las empresas

En la actualidad las empresas de alto rendimiento recurren a la externalización de los procesos de negocio, con el fin de adaptarse a la necesidad de cambio del mercado actual. Son muchas las ventajas que se pueden obtener de la utilización de esta herramienta, desde el punto de vista jurídico hasta el económico. Sin embargo, como todo proceso, no queda exento de la existencia de aspectos negativos en su utilización. [4]

A continuación, citaremos las principales ventajas y desventajas, al externalizar los procesos de una empresa:



VENTAJAS	DESVENTAJAS
<ul style="list-style-type: none">• Permite a la empresa centrar sus esfuerzos en su negocio principal, lo que conlleva a una mayor rentabilidad y aumento de calidad.• Reducción y manejabilidad de costes: se reduce la inversión en planta y equipo.• Conversión de costes variables en fijos, debido a que una parte de las operaciones de la empresa queda estipulada por un Contrato de Outsourcing con un precio fijo, lo que permite a la compañía estabilizar sus precios.• Reducción del riesgo operacional: la empresa cliente transfiere el riesgo en todas las operaciones externalizadas.• Aprovechamiento del conocimiento y capacidad del proveedor.	<ul style="list-style-type: none">• Se proporciona información confidencial a otra empresa sobre los productos y procesos, pudiendo ser utilizado para empezar una compañía propia que se convierta en competidor directo.• Se pierde el control sobre la producción o procesos.• Una empresa de BPO puede estar asociada con varias empresas a la vez, lo que puede traducirse en una disminución de los tiempos de respuesta y/o calidad en el servicio contratado.• La empresa va perdiendo el contacto con las nuevas tecnologías, que sirven de oportunidad para poder innovar en los productos y procesos.

Tabla 1: Ventajas y desventajas del BPO para las empresas.

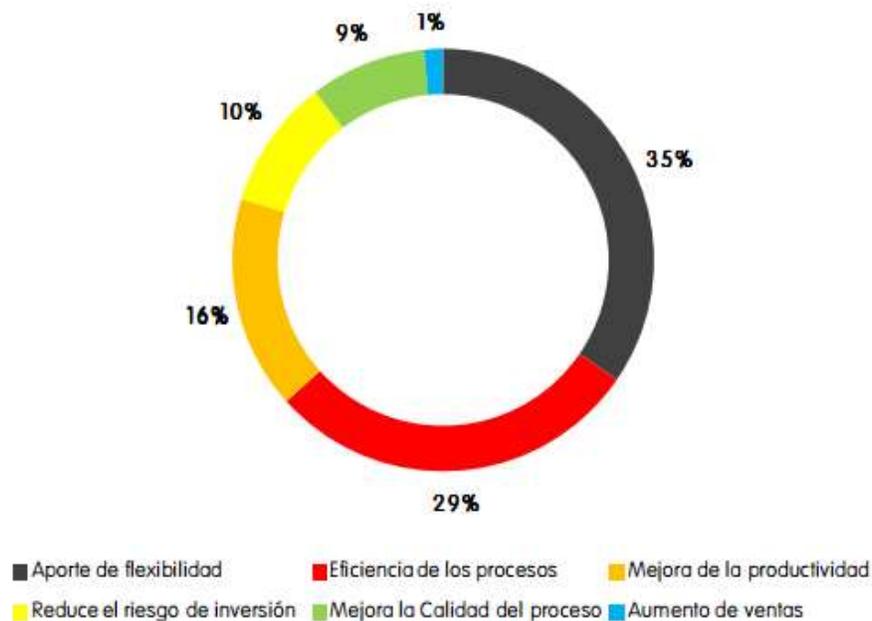


Gráfico 4: Ventajas del uso de la Externalización en las empresas [2].

II.1.5 Everis BPO

BPO es un área de la empresa Everis que presta servicios de externalización de procesos de negocios a organizaciones de todo el mundo, aportando el talento y la excelencia de los profesionales que conforman la empresa (un 70 % de los profesionales de Everis son titulados superiores).

Es una de las divisiones de la compañía donde se pronostica un mayor crecimiento, debido a los avances exponenciales que ha tenido en los últimos años (ver Gráfico 5), gracias a las herramientas que utiliza para realizar estos contratos y que le permite marcar un punto diferencial con la competencia, entre las que podemos citar:

- **Formación:** poner en marcha sistemas de aprendizaje organizativo, internos y externos, que eviten errores y que permita incorporar las mejores prácticas.
- **Tecnología:** implantación de herramientas *Business Process Management* que permitan obtener datos sobre los procesos y medir su funcionamiento.

- **Un buen contrato:** reflejar en el contrato las fases de transferencia y devolución del servicio.



Gráfico 5: Evolución de ventas de Everis BPO [5] .

II.1.6 Provisión del servicio

La provisión de servicio es un proceso de negocio en que se encuentran involucradas diferentes unidades de una organización con el fin de generar valor para cliente.

Enfocado en el sector de las telecomunicaciones, es el proceso de negocio que engloba todas las tareas que conducen a proporcionar, modificar o eliminar servicios de telecomunicación a clientes, en base a sus pedidos realizados, es decir, comprende desde que el cliente manifiesta deseo de un servicio (alta), hasta la modificación de las características de la prestación que tienen, hasta el traslado, baja, instalación y facturación del mismo. [6]

A muy groso modo, se puede decir que este proceso comprende 6 fases, como se puede visualizar en la figura #2.

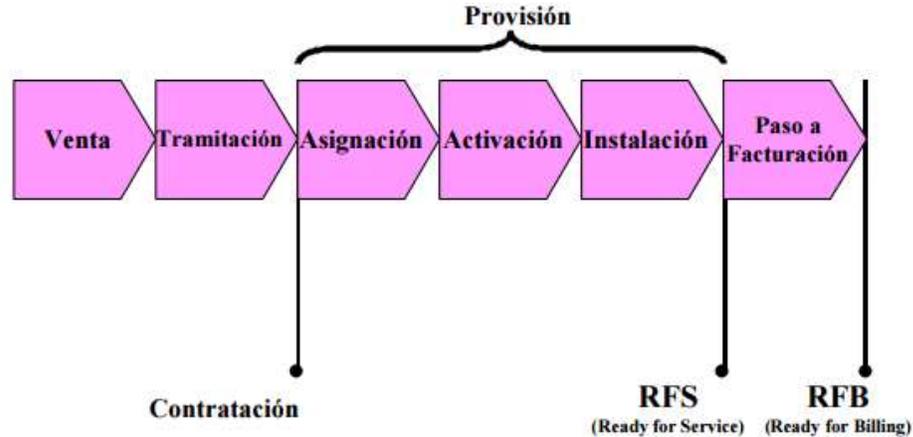


Figura 2: El proceso de provisión de servicio a un cliente [6].

Como todo proceso, comienza debido a la necesidad ocasionada en un cliente, que origina una actividad comercial por parte de la empresa y es donde se produce la venta del servicio. Posteriormente, se recopilan los datos del cliente mediante el proceso de tramitación, y se cierra el proceso de la venta mediante la contratación, y es ahí donde comienza la provisión del servicio, en la que se comprenden tres tipos de actividades técnicas:

- **Asignación:** separar los recursos físicos y lógicos para establecer y configurar el servicio.
- **Activación:** configuración de centrales mediante la ejecución de comandos.
- **Instalación:** actividades técnicas manuales realizadas en las centrales, planta exterior o dependencias del cliente.

Una vez finalizadas estas etapas se considera que el servicio se encuentra preparado para ser utilizado por el cliente, denominado *Ready For Service* (RFS). Sin embargo, hay que tener en cuenta que se debe cumplimentar un paso más, muy importante en este proceso, denominado facturación. La facturación permite mediante diversos sistemas programados comenzar a facturar un servicio a un cliente. Este punto final es el denominado *Ready For Billing*.

II.2 Sistemas y Bases de Datos

Los sistemas de gestión de bases de datos tienen como objetivo: almacenar los datos en un computador, de forma segura y consistente, para permitir las solicitudes de acceso a los datos formuladas por los usuarios. Estos sistemas permiten la implementación de bases de datos utilizando algún modelo, con el fin de satisfacer demandas particulares de los usuarios.

Para la definición de la estructura de la base de datos (BD), realizada por el diseñador según un modelo de datos (MD), así como para la utilización de ella por los usuarios, son necesarios lenguajes de comunicación con los SGBD.[7]

Estos lenguajes deben así permitir las siguientes operaciones:

- Definir la Estructura de la base de datos (BD).
- Crear las bases de datos necesarias.
- Eliminar bases de datos existentes.
- Reestructurar las bases de datos existentes.
- Consultar acerca de la estructura de las bases de datos existentes.
- Manipular los Datos Almacenados.
- Insertar de datos en las bases de datos.
- Modificar datos de las bases de datos.
- Consultar datos de las bases de datos.

II.2.1.- Lenguajes

Generalmente, los SGBD utilizan algún lenguaje específico, de forma que sólo a través de él se puede interactuar con el sistema para realizar alguna de las operaciones expuestas. La estructura del lenguaje depende directamente del modelo de datos que implementa el SGBD.



El lenguaje como tal sólo es parte de un conjunto de herramientas que permiten la implementación de sistemas de información computacionales.

Los lenguajes de SGBD se encargan de proveer un protocolo de comunicaciones con los SGBD, de forma que cualquier usuario tenga acceso al SGBD (a través de una red, por ejemplo) y pueda realizar transacciones con él. Se debe notar que los usuarios pueden ser personas, aplicaciones cliente, o incluso otros SGBD.

De esta manera, cuando un usuario desea comunicarse con un SGBD, puede hacerlo sólo a través del lenguaje del SGBD, ya sea utilizando una interfaz de línea de comandos, que permite realizar cualquier operación con el lenguaje; o una aplicación cliente específica, desarrollada para satisfacer necesidades particulares de información, y que utiliza el mismo lenguaje.

En el caso de los SGBD Relacionales (basados en el Modelo Relacional de datos), el lenguaje utilizado es SQL (*Structured Query Language*).

II.2.1.1.- Elementos de un lenguaje

Los lenguajes de SGBD deben permitir realizar todas las operaciones antes mencionadas sobre las bases de datos y los datos que ellas contienen:

- Lenguaje de Definición de Datos (LDD): Permite definir y manipular la estructura de bases de datos almacenadas en el SGBD.
- Lenguaje de Manipulación de Datos (LMD): Permite ingresar, modificar y eliminar datos de las bases de datos almacenadas en el SGBD, realizar y controlar las transacciones con el SGBD.

II.2.1.2.- El Lenguaje SQL y el Modelo Relacional

La función del lenguaje SQL es la de soportar la definición, manipulación y control de los datos en una base de datos relacional. Desde el punto de vista de este lenguaje



una base de datos relacional es un conjunto de tablas, donde cada tabla es un conjunto de filas y columnas.

II.2.1.2.1.- SQL como Lenguaje

Primero que nada debe quedar claro que SQL no es un lenguaje de programación como C o Pascal, ya que no posee sentencias de selección o iteración (IF, THEN, FOR, WHILE, etc.). Por lo tanto, cuando se desean utilizar los datos de la base de datos (BD) para realizar algún proceso, se debe programar en otro lenguaje (lenguaje anfitrión), desde el cual se hacen llamadas a sentencias SQL (SQL incrustado) cuando se desea interactuar con la BD.

SQL se aplica de forma natural al problema de implementar BD Relacionales, pero como consecuencia de esto es poco estructurado y utiliza muchas palabras superfluas (que podrían omitirse quitando claridad a las sentencias).

Finalmente SQL se adapta bien a la administración de datos en un entorno de red, especialmente en arquitecturas de cliente/servidor. Esta proporciona muchas herramientas y es compatible con muchos lenguajes de programación.

II.2.1.2.2.- SQL y los SGBD

SQL (*Structured Query Language*) es mucho más que un lenguaje de consulta de BD, ya que permite mediante el uso del Modelo Relacional de datos:

- Definir la estructura de los datos.
- Recuperar y manipular datos.
- Administrar y controlar el acceso a los datos.
- Compartir datos de forma concurrente.
- Asegurar su integridad.

En consecuencia, SQL permite hacer efectivos los objetivos de los SGBDR (Sistemas de Gestión de Bases de Datos Relacional), permitiendo a los usuarios utilizar todo el potencial de los SGBDR.

- **SQL: LDD y LMD**

SQL posee sentencias tanto para definición de la estructura de los datos, como para la manipulación de ellos. Así, SQL provee sentencias para ambos tipos de tareas.

- **SQL y el Modelo Relacional (MR)**

El Modelo Relacional se compone, formalmente, de las siguientes partes:

- Parte Estática: Dominios, atributos, relaciones, reglas de Integridad.
- Parte Dinámica: Algebra Relacional.

El lenguaje SQL permite representar casi todos los elementos del MR, salvo los conceptos de dominio (que no se soporta) y de relación (que se soporta de forma restringida).

Cabe destacar que la manipulación de datos provista por SQL, es más poderosa que la definida originalmente en el MR, mediante el álgebra relacional. Esto ha motivado que ésta haya sido extendida para representar las posibilidades de SQL.

- **Dominios**

En SQL, sin embargo, el propósito de los dominios es permitir la especificación de un tipo de dato, tal como CHAR(5). En el cual, los tipos de datos se definen sólo una vez, permitiendo de este modo que dicha especificación sea usada por columnas pertenecientes a muchas tablas.

Las diferencias más importantes entre los dominios SQL y dominios relacionales son:

- Los dominios SQL son especificaciones sintácticas; no son tipos de datos definidos por el usuario.
- Los dominios SQL no siempre se usan, ya que las columnas de las tablas pueden definirse directamente en términos de tipos de datos soportados por el sistema.
- Los dominios SQL no soportan definiciones de dominios en términos de otros dominios, ya que se definen siempre en términos de tipos de datos del sistema (no hay herencia).
- Los dominios SQL no permiten chequeos, ya que los únicos requerimientos para hacer comparaciones es que los elementos sean del mismo tipo de datos.
- SQL no permite que los usuarios definan operaciones que se apliquen a los dominios.
- SQL no hace una clara distinción entre un dominio SQL y la representación de ese dominio en términos de tipos de datos soportados por el sistema.

En resumen, los dominios SQL no soportan el concepto fundamental de dominio relacional: el dominio de valores verdaderos definidos por el usuario.

- **Relaciones y tablas base**

Las tablas de SQL no son relaciones, ya que no cumplen con las propiedades recién mencionadas. La forma en que se representa una relación en SQL es mediante una tabla base, las cuales tienen las siguientes características anómalas:

- Las tablas base de SQL permiten filas repetidas, a no ser que se defina un mecanismo que lo evite.
- Las tablas SQL tienen un orden de columnas, de izquierda a derecha, y de filas, de arriba a abajo.

- **Claves Candidatas**

Las claves candidatas en SQL se declaran como: UNIQUE (*column-commalist*) ó PRIMARY KEY (*column-commalist*), donde “*column-commalist*” no debe ser vacío.

El MR (Modelo Relacional) asegura que toda relación tiene al menos una clave candidata.

- **Facilidades de Implementación de BD (Bases de datos) Relacionales con SQL**

Para ofrecer funcionalidades que permitan a los diseñadores de sistemas de bases de datos, los SGBDR ofrecen extensiones del SQL, basadas en *triggers*, procedimientos almacenados y cursores.

Además SQL con su característica de ser un lenguaje dual, le entrega a los usuarios finales la posibilidad de que sean ellos mismos los que realicen consultas a la base de datos.

- **Dualidad del Lenguaje**

Los comandos del lenguaje SQL pueden ser ejecutados interactivamente o bien formando parte de un programa de aplicación. En este último caso, las sentencias de SQL están insertas dentro del código fuente, mezcladas con las sentencias del lenguaje anfitrión.

Existen diferencias entre las sentencias SQL declaradas interactivamente y su contraparte embebida.

Además, algunas sentencias de SQL embebido no pueden usarse interactivamente y viceversa.



Para realizar eficientemente programas de aplicación, los SGBDR comerciales ofrecen una extensión de SQL, que consiste en agregar sentencias procedurales tradicionales, tales como IF-THEN-ELSE, FOR, WHILE, etc.

II.2.2 SQL Server:

Es un sistema de administración y análisis de bases de datos relacionales de Microsoft para soluciones de comercio electrónico, línea de negocio y almacenamiento de datos.

Microsoft SQL Server se basa en funciones críticas que proporcionan un rendimiento, una disponibilidad y una facilidad de uso innovadores para las aplicaciones más importantes.

El lenguaje de desarrollo utilizado, ya sea por línea de comandos o mediante la interfaz gráfica de Management Studio, es *Transact-SQL* (TSQL), una implementación del estándar ANSI del lenguaje SQL, utilizado para manipular y recuperar datos, crear tablas y definir relaciones entre ellas.

Transact-SQL es el principal medio de interacción con el Servidor, el cual permite realizar las operaciones claves en SQL Server, incluyendo la creación y modificación de esquemas de base de datos, inserción y modificación de datos en la base de datos, así como la administración del servidor como tal. Esto se realiza mediante el envío de sentencias en T-SQL y declaraciones que son procesadas por el servidor y los resultados (o errores) regresan a la aplicación cliente.

Dentro de las capacidades y herramientas básicas de SQL Server, las que más se utilizarán durante el desarrollo del proyecto son las siguientes:

- Creación de nuevas Bases de Datos
- Creación de tablas fijas y temporales
- Definición de tipos de datos para cada columna de cada tabla
- Creación de vistas

- Creación de Procesos almacenados
- Consultas distribuidas

II.2.3 Lotus Notes:

Lotus notes es un sistema desarrollado por la empresa IBM que actúa como cliente servidor incluyendo correo electrónico. Cuando se refiere a la parte del servidor, se hace referencia a el nombre de Lotus Dominio, y cuándo se hace referencia al cliente, se llama Lotus Notes. En la parte de servidor, IBM proporciona versiones para las diferentes plataformas que existen (Windows, HP-ux, I5OS, etc.). Mientras que de lado cliente (Lotus Notes) cuenta con versiones para Windows y Mac. En general, lo que este sistema permite es el manejo de correo electrónico, calendarios y manejo de juntas. Otra de las ventajas que da Lotus Notes es la facilidad para incluir bases de datos, documentales, los procedimientos, manuales y hasta foros de discusión que se pudieran hacer dentro del área de trabajo de los mismos empleados. A continuación se puede observar su menú inicial:



Figura 3: Menú inicial del IBM Lotus Note.

II.2.5 Access:

Microsoft Access es un sistema de gestión de bases de datos incluido en Microsoft Office. Está pensado en recopilar datos de otras utilidades (Excel, SharePoint, etc.) y manejarlos por medio de las consultas e informes.

II.2.6 Gestión de Circuitos (GESCIR):

En el Proyecto se manejan varias bases de datos en Access, entre ellas está la de Gestión de Circuitos (GESCIR), cuyo menú de inicio se ve en la figura 5, en la que se va recopilando la información sobre cada línea telefónica y circuito que se vaya pidiendo para cada ítem, así como sus estados, fechas de solicitudes, incidencias, observaciones, etc.

Un proceso de carga automática importa un resumen de datos de la base de datos de GESCIR a la base de datos de SQL Server del servidor.

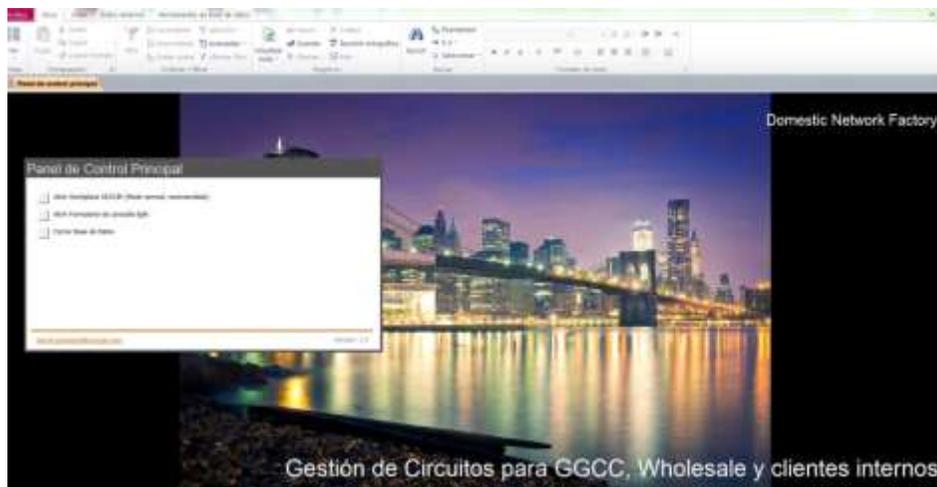


Figura 5: Base de datos GESCIR.

II.2.7 HP Inventory:

Es un inventario donde se almacena información de cada equipo instalado y configurado durante todo el proyecto, así como además se puede probar las

configuraciones DNS, SNMP e ICMP de los equipos, como se puede ver a continuación:



Figura 6: Opciones asociadas a cada equipo en *HP Inventory*.

Para acceder a este inventario se debe estar conectado a la VPN del cliente y acceder a la Web de *HP Inventory*. Una extracción diaria de este Inventario es enviada al buzón de correo de nuestro proyecto por parte del personal del cliente.

Una vez realizada la instalación del equipo, para dar de alta a dicho equipo en *HP Inventory* se deben llenar datos del cliente, sede, datos de configuración y especificaciones del equipo, indicando datos técnicos para cada interfaz configurada. En la siguiente figura se observa la opción para crear un equipo en *HP Inventory*:



Figura 7: Alta de un equipo en *HP Inventory*.

II.2.8 NA *Inventory*:

HP *Network Automation (NA) Inventory*, al igual que el inventario mencionado anteriormente, recopila información sobre los equipos que hayan sido instalados en el proyecto.

Network Automation es un componente de la solución de gestión de red de HP que proporciona un enfoque integral y automatizado a través del dominio de gestión de redes. Este software automatiza el ciclo de vida operativo completo de los dispositivos de red, desde el aprovisionamiento hasta la gestión de cambios basados en políticas, el cumplimiento y la administración de la seguridad.[8]

Ventajas del uso de HP NA:

- Reduce los costes mediante la automatización de las tareas manuales de configuración y cumplimiento que consumen mucho tiempo.
- Ayuda a superar auditorías y requisitos de cumplimiento con la aplicación de políticas proactivas e informes de auditoría y cumplimiento.
- Mejora la seguridad de la red mediante el reconocimiento y la solución de vulnerabilidades de la seguridad antes de que impacten en la red
- Aumenta la estabilidad de la red y el tiempo de actividad mediante la prevención de las inconsistencias y las configuraciones erróneas
- Aprovecha la automatización impulsada por los procesos para ofrecer integraciones de aplicaciones.

II.2.9 Tacacs

Terminal Access Controller Access Control System (TACACS), traducido como ‘sistema de control de acceso mediante control del acceso desde terminales’ es un protocolo de autenticación remota, propietario de cisco, que se usa para comunicarse con un servidor de autenticación comúnmente usado en redes Unix. TACACS



permite a un servidor de acceso remoto comunicarse con un servidor de autenticación para determinar si el usuario tiene acceso a la red. TACACS está documentado en el RFC 1492.

En el proyecto se usa una autenticación con usuarios de *Tacacs* para validar si un equipo específico ha sido dado de alta correctamente y tiene acceso a la red.

II.2.10 Confluence

Es un software de colaboración en equipo. Está escrito en Java y utilizado principalmente en empresas, está desarrollado y comercializado por *Atlassian*. *Confluence* se vende tanto como software de uso local como solución de servidor.

Confluence es considerado como una empresa emergente en software social, bastante popular en entornos corporativos, fácil de instalar y de utilizar. En varios proyectos se usa *confluence* de diferentes maneras, ya sea como herramienta de control de la ocupación de recursos con el tiempo y agendaciones, así como repositorios de documentaciones y datos. En el proyecto de Orange de BPO, se utiliza un repositorio en *confluence* donde se guarda un documento Excel de plantilla de '*ReadyforService*' (RFS) por cada instalación realizada con éxito.

II.3 Términos relacionados al Proyecto

II.3.1 Router

Es un dispositivo que proporciona conectividad a nivel de red. Su función principal consiste en enviar o encaminar paquetes de datos de una red a otra, es decir, interconectar subredes, entendiendo por subred un conjunto de máquinas IP que se pueden comunicar sin la intervención de un encaminador.

El funcionamiento básico de un enrutador o encaminador, como se deduce de su nombre, consiste en enviar los paquetes de red por el camino o ruta más adecuada en cada momento. Para ello almacena los paquetes recibidos y procesa la información de



origen y destino que poseen. Con arreglo a esta información reenvía los paquetes a otro encaminador o bien al anfitrión final, en una actividad que se denomina 'encaminamiento'.

Los equipos que actualmente se le suelen vender al consumidor como enrutadores no son simplemente eso, sino que son los llamados Equipos locales del cliente (CPE). Los CPE están formados por un módem, un enrutador, un conmutador y opcionalmente un punto de acceso *WiFi*.

II.3.2 Velocidad de sincronismo

Se refiere a la velocidad de subida y bajada del equipo, viene dada en bits por segundos. La velocidad de envío y recepción de datos de cada puerto, o interfaz del equipo no siempre es igual, en los *router* simétricos (SHDSL) las velocidades de subida y baja son teóricamente iguales.

II.3.3 DomainNameService DNS

El DNS (*Domain Name Service*) es un sistema de nombres que permite traducir de nombre de dominio a dirección IP y vice-versa. Aunque Internet sólo funciona en base a direcciones IP, el DNS permite que los humanos usemos nombres de dominio que son bastante más simples de recordar (pero que también pueden causar muchos conflictos, puesto que los nombres son activos valiosos en algunos casos).

El sistema de nombres de dominios en Internet es un sistema distribuido, jerárquico, replicado y tolerante a fallas. Aunque parece muy difícil lograr todos esos objetivos, la solución no es tan compleja en realidad. El punto central se basa en un árbol que define la jerarquía entre los dominios y los sub-dominios.

Inicialmente, el DNS nació de la necesidad de recordar fácilmente los nombres de todos los servidores conectados a Internet. De no existir los servidores DNS los usuarios tendrían que escribir la dirección IP del sitio web en lugar de escribir la URL



de este lo cual generaría confusiones y la navegación en internet se tornaría muy complicada para los usuarios.

II.3.4 Ancho de Banda

El ancho de banda es la cantidad de información o de datos que se puede enviar a través de una conexión de red en un período de tiempo dado. El ancho de banda se indica generalmente en bites por segundo (BPS), *kilobites* por segundo (kbps), o *megabites* por segundo (mps).

El ancho de banda a menudo se utiliza como sinónimo para la tasa de transferencia de datos - la cantidad de datos que se puedan llevar de un punto a otro en un período dado (generalmente un segundo).

II.3.5 Servidor

Un servidor es una aplicación en ejecución (software) capaz de atender las peticiones de un cliente y devolverle una respuesta en concordancia. Los servidores se pueden ejecutar en cualquier tipo de computadora. En la mayoría de los casos una misma computadora puede proveer múltiples servicios y tener varios servidores en funcionamiento. La ventaja de montar un servidor en computadoras dedicadas es la seguridad. Por esta razón la mayoría de los servidores son procesos diseñados de forma que puedan funcionar en computadoras de propósito específico.

Los servidores operan a través de una arquitectura cliente-servidor. Los servidores son programas de computadora en ejecución que atienden las peticiones de otros programas, los clientes. Por tanto, el servidor realiza otras tareas para beneficio de los clientes. Ofrece a los clientes la posibilidad de compartir datos, información y recursos de hardware y software. Los clientes usualmente se conectan al servidor a través de la red pero también pueden acceder a él a través de la computadora donde está funcionando.

Comúnmente los servidores proveen servicios esenciales dentro de una red, ya sea para usuarios privados dentro de una organización o compañía, o para usuarios públicos a través de Internet. Los tipos de servidores más comunes son servidor de base de datos, servidor de archivos, servidor de correo, servidor de impresión, servidor web, servidor de juego, y servidor de aplicaciones.

II.3.6 Interfaces

Una interfaz de *router* suministra la conexión física entre el *router* y un tipo de medio físico de la red. Las interfaces a menudo se denominan puertos. Los puertos incorporados se designan por su tipo de conexión seguido de un número. [9]

En el caso de los enrutadores encontramos dos tipos de interfaces:

- Interfaces encaminadas: son interfaces de nivel 3, accesibles por IP. Cada una se corresponde con una dirección subred distinta. Se distinguen a su vez dos subtipos:
 - Interfaces físicas: aquellas accesibles directamente por IP.
 - Interfaces virtuales: aquellas que se corresponden con una VLAN o un CV. Si dicha interfaz se corresponde con una única VLAN se denomina *Switch Virtual Interfaz* (SVI), mientras que si se corresponde con un enlace *trunk* o con un CV, actúan como subinterfaces.
- Interfaces conmutadas: se trata de interfaces de nivel 2 accesibles solo por el módulo de conmutamiento. Reciben el nombre de puertos de conmutador. Las hay de dos tipos:
 - Puertos de acceso: soportan únicamente tráfico de una VLAN.
 - Puertos *trunk*: soportan tráfico de varias VLANs distintas.



Estas posibilidades de configuración están únicamente disponibles en los equipos modulares, ya que en los de configuración fija, los puertos de un enrutador actúan siempre como interfaces encaminadas, mientras que los puertos de un conmutador como interfaces conmutadas. Además, la única posible ambigüedad en los equipos configurables se da en los módulos de conmutamiento, donde los puertos pueden actuar de las dos maneras, dependiendo de los intereses del usuario.

II.3.7 Equipo Local del Cliente(CPE)

El CPE (Equipo Local del Cliente) es un término de telecomunicaciones usado tanto en interiores como en exteriores para originar, encaminar o terminar una comunicación. El equipo puede proveer una combinación de servicios incluyendo datos, voz, video y un host de aplicaciones multimedia interactivos.

Históricamente, este término se refería al equipamiento situado en el extremo de la línea telefónica del usuario, y normalmente era propiedad de la compañía de teléfono. Hoy en día, sin embargo, prácticamente cualquier equipo de usuario final se puede denominar *Customer Premises Equipment* (CPE), y puede ser propiedad tanto del usuario como del proveedor. Pero aunque puede ser propiedad cualquiera de los dos, el CPE suele ser del usuario y se sitúa en la conexión eléctrica del mismo o directamente en un enchufe.

Uno de los tipos de CPE es el de ADSL. Para contar con el servicio de internet ADSL el usuario debe contar con lo siguiente:

- Que su proveedor de internet cuente con la tecnología necesaria para ofrecer el servicio.
- Una línea telefónica.
- Un CPE (*Customer Premises Equipment*) que puede ser propiedad del cliente o suministrado por su proveedor de internet.
- Un computador con puerto USB o LAN.



II.3.8 Página Web

Una página Web es un documento electrónico que puede soportar texto, imágenes y/o música, que se encuentra identificado mediante un nombre de dominio.

Para desarrollarlas se utilizan lenguajes de programación que los navegadores (Chrome, Mozilla, Internet Explorer y Safari) puedan interpretar posteriormente. Entre los lenguajes más utilizados podemos citar: HTML, PHP, ASP, JSP o RUBY.

Es necesario alojar los portales web en un servidor (ordenador conectado constantemente a una intranet privada, que permite tener accesible la página web para los usuarios a cualquier hora del día), el cual podemos acceder con el uso de programas denominados clientes FTP, los cuales permiten conectar el ordenador personal donde se desarrolló la página web al servidor donde la alojaremos. Esta conexión del ordenador personal al servidor web, alojando allí la página desarrollada, permite que desde un navegador web se pueda acceder a la página por medio de peticiones al servidor web (véase figura 8).

En la actualidad, con el aumento exponencial del uso del internet en la vida cotidiana, las mayoría de las empresas cuentan con su portal web, para darse a conocer al mundo, con el fin de introducirse en el mercado laboral, obtener nuevos clientes y mejorar sus resultados económicos.

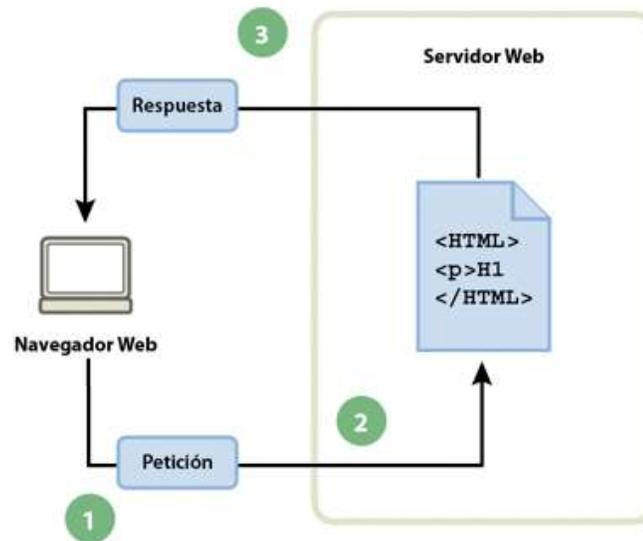


Figura 8: Relación servidor y navegador web [10].

II.4 Lenguajes de Programación

II.4.1 Vb.net

Visual Basic .Net es un lenguaje de programación eficaz orientado a objetos, ya que incorpora numerosas características nuevas y mejoradas con respecto al Visual Basic, como por ejemplo: herencia, interfaces, sobrecarga, control de excepciones y subprocesamiento. Posee un conjunto de herramientas de desarrollo para la construcción de aplicaciones Web ASP, servicios Web XML, aplicaciones para escritorio y aplicaciones móviles, que permite simplificar su desarrollo en un entorno sumamente distribuido: Internet. Posee interoperabilidad entre lenguajes, mayor seguridad y compatibilidad entre versiones. Se considera una evolución de Visual Basic que ha sido implementada sobre el *framework* .NET.

Se suele utilizar como entorno de desarrollo integrado el Microsoft Visual Studio en alguna de las versiones disponibles en el mercado, aunque existen otras alternativas como *SharpDevelop*, que es un software libre. Es considerado como la competencia

de la plataforma Java y PHP por parte de Microsoft, a causa del exponencial crecimiento del mercado de negocios en entornos web.

II.4.2 SSH

En la actualidad, la mayoría de conexiones a internet se realizan mediante el protocolo http, que se caracteriza por transferir información en texto plano sin cifrar, por lo que se puede interceptar el tráfico fácilmente.

SSH (*Secure Shell*, en español intérprete de ordenes seguro) es un protocolo que permite acceder a máquinas remotas de manera segura a través de la red, lo que permite remplazar los métodos antiguos debido a su vulnerabilidad. Este protocolo hace uso de la arquitectura de cliente-servidor como se observa en la figura 9.



Figura 9: Funcionamiento del protocolo SSH [11].

El funcionamiento del protocolo se puede resumir en los siguientes pasos:

1. El cliente inicia una conexión TCP sobre el puerto 22 de servicio (se puede modificar). El cliente puede verificar que está realizando la conexión al mismo servidor anterior.
2. El cliente y el servidor se sincronizan para utilizar la misma versión de protocolo y algoritmo de cifrado para intercambiar información.



3. El servidor posee una clave privada y una pública que es enviada al cliente, mediante encriptación de 128 bits, lo que lo hace robusto para descifrar y leer.
4. El cliente al recibir la clave enviada por el servidor la compara con la que posee almacenada y así verifica su autenticidad. El protocolo SSH exige que el cliente la confirme la primera vez.
5. El cliente genera una clave de sesión aleatoria, creando un mensaje que contiene esa clave y el algoritmo seleccionado para la encriptación de la información. Toda esa información es enviada al servidor haciendo uso de la clave pública que envió en un paso anterior de forma cifrada.
6. Si todo es correcto, el cliente queda autenticado, iniciando la sesión para comunicarse con el servidor.

II.4.3 ASP

ASP (*Active Server Pages*) es un lenguaje de scripting del lado del servidor creado por Microsoft que forma parte del IIS 3.0 (*Internet Information Server*), y permite usar diferentes scripts y crear contenido HTML para mostrar páginas generadas dinámicamente, aplicaciones web y servicios web XML. El principio de esta tecnología es el VBScript, lenguaje de programación interpretado y no compilado, es decir, se ejecuta del lado del servidor.

Posteriormente, con el surgimiento del .NET nació el ASP.net que contiene archivos de extensión .ASPX, que soporta código HTML y código de servidor para soportar una transición más sencilla desde el ASP clásico. Sin embargo, lo ideal es trabajar con dos archivos por separado, que es lo que hace Visual Studio por defecto.

El funcionamiento de ASP se puede resumir en los siguientes pasos:

1. El usuario solicita un documento ASP desde su navegador. La comunicación hacia el servidor se da por medio de la web.



2. El Servidor procesa los contenidos del ASP para generar contenidos en forma dinámica y temporal para el usuario.
3. Los archivos ASP son un conjunto de instrucciones HTML, Scripts ASP en Vbscript u otros lenguajes, así como conexiones a bases de datos o componentes.

Por resultado, obtiene un documento estático como cualquier otro HTML que fue generado en el servidor gracias al ASP.

II.4.4 Javascript

JavaScript es un lenguaje de programación utilizado para crear páginas web dinámicas, es decir, que incorporan efectos en los elementos que la componen (texto que aparece y desaparece, ventanas con mensajes de avisos, etc.). Es un lenguaje interpretado, es decir, se pueden probar en cualquier navegador sin necesidad de ser compilados por algún programa o proceso intermedio.

Se encuentra presente prácticamente en todos los ámbitos desde sistemas operativos, desarrollo móvil, servidores de internet hasta bases de datos y plataformas de juegos. Como ventajas de su utilización se encuentran:

- Es un lenguaje que funciona del lado del cliente, por lo que no requiere compilación.
- Es un lenguaje muy sencillo de aprender.
- El lenguaje de scripting es seguro y fiable.
- Fácil de integrar.
- Puede ser ejecutado en cualquier página sin necesidad de instalar otro programa para poder visualizarlo.
- Puede ser utilizado para crear pequeños programas insertados en una página web hasta programas grandes y más complejos orientados a objetos.
- Permite mejorar la interfaz de usuario mediante la creación de páginas web dinámicas, con elementos interactivos y gran cantidad de efectos visuales.



- Es soportado por la mayoría de navegadores como: Internet Explorer, Opera, Mozilla Firefox, Opera, etc.
- Es un lenguaje de programación muy liviano por lo que utiliza muy poca memoria, por lo que mantiene un tiempo de descarga rápido en la página web.

II.4.5 HTML

HTML es el lenguaje que se emplea para desarrollar el contenido de páginas web. Está compuesto por una serie de etiquetas que el navegador interpreta y da forma en la pantalla, mediante la marcación de elementos, desde texto hasta imágenes, hipervínculos, listas, videos, etc, es decir, cuando se desea añadir un elemento externo a una página, no se encaja directamente, sino se realiza mediante texto, especificando su ubicación. Esto se traduce en mejor rendimiento, ya que la programación de la página web solo contiene texto, y es el navegador web el encargado de unir todos los elementos para la visualización final al usuario. Entre las principales ventajas de la utilización de este lenguaje tenemos:

- Es un lenguaje bastante amigable de utilizar, por lo que resulta muy fácil su aprendizaje, interpretación y uso.
- Para programar una página en HTML solo es necesario tener un editor de texto plano y guardar el archivo con extensión “.html”.
- Consta de etiquetas (aunque bastante limitadas) para poder proporcionarle estilos al texto en una página web.
- Es el lenguaje más extendido, todos los navegadores lo admiten.

Existen muchas aplicaciones y editores de páginas web que generan el código HTML automáticamente.

II.4.6 CSS

La Hoja de estilo en cascada o CSS, siglas en inglés de *Cascading Style Sheets*, es un lenguaje que utilizado para proporcionar el estilo y la presentación deseada a un

documento electrónico creando previamente en HTML o XML. Además, permite separar la estructura del contenido (párrafo, título, tabla, lista de elementos, etc.) de la presentación (color, tamaño, estilo de letras), para mejor manejabilidad de la programación del portal web. Entre las principales ventajas de utilizar este lenguaje de programación, las cuales también se observan en la figura 10, se pueden citar:

- Al dividir el contenido y la apariencia de una página web permite generar archivos más ligeros, lo que permite disminuir los tiempos de carga del sitio al servidor, permitiendo mantener en caché nuestra hoja de estilos.
- Se establecen mismos estilos para varios elementos con el mismo comportamiento y/o el mismo documento CSS puede servir para varios documentos HTML, lo que permite agilizar el ancho de banda de conexión.
- Se puede complementar los estilos con otros lenguajes de programación más potentes como JavaScript para conseguir efectos dinámicos.
- Se permite establecer una hoja de estilo por diferente navegador.
- Contiene herramientas más potentes que las que proporciona el lenguaje HTML.
- La utilización de CSS nos permite tener un código más limpio y estructurado, y agiliza la tarea de modificación del código.

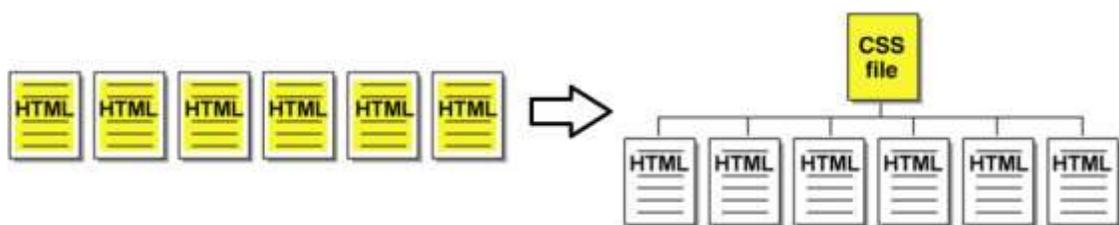


Figura 10: Ventajas del uso de CSS en el desarrollo de páginas web [10].

CAPÍTULO III

Metodología

Una vez definidas las bases teóricas del trabajo, a continuación se presenta la metodología que le da forma, y las actividades de cada una de las fases que la componen.

Fase I: Proceso de investigación y preparación.

- Se realizará la petición de requerimientos globales y específicos, acordando tipos de evaluaciones, y muestreo de resultados.
- Se analizarán los procesos de evaluaciones que se van a realizar, aprobando el método en que se evaluará cada validación.
- Se estudiarán los lenguajes de programación y herramientas requeridas para la elaboración de la herramienta.
- Se plantearán validaciones estimadas de tiempo.

Fase II: Página de control, gestión y soporte de PDS dentro de la Web

- Se añadirá una página a la web de *PdS* que permita soporte y control de los estados de las instalaciones, donde los técnicos y personal de *PdS* puedan introducir los datos correspondientes a cada equipo e instalación que serán nuestros datos de entrada al proceso de validación.
- Modificación del código de la web de PDS.
- Se crearán nuevos campos y tablas en la BBDD de la web de *PdS* donde se cargarán los datos de la nueva página.



Fase III: Modelado de Datos de entrada al proyecto.

- Se crearán las tablas correspondientes al validador e inventario en la BBDD de desarrollos.
- Creación de un procedimiento de exportación de los datos correspondientes a las nuevas instalaciones finalizadas desde las tablas de la BBDD de la Web hacia la tabla del validador en la BBDD de desarrollo.
- Se añadirán los nuevos campos necesarios a los procesos de carga de WOM (Lotus Notes) y GESCIR (Access) desde sus bases de datos a nuestra BBDD de desarrollo. Ya existen procesos de carga internos del proyecto, sólo se le agregarán los campos nuevos que sean necesarios para inventariar.

Fase IV: Procesos de carga automáticos de los Inventarios.

- Se desarrollarán tablas en la BBDD donde se importarán los datos de los inventarios de HP y NA.
- Se desarrollará un automatismo de carga del Inventario de NA desde su respectiva web.
- Se desarrollará un automatismo de carga del Inventario de HP desde una extracción Excel que se enviará al buzón de correo de nuestro servidor.

Fase V: Desarrollo del validador e inventariado

- Desarrollo de las vistas desde la BBDD en SQL server que crucen todas las tablas de datos de entrada y agrupen los datos necesarios para la validación e inventariado.
- Desarrollo del proceso de inventariado para aquellos registros correspondientes a nuevas instalaciones que aún no se han inventariado.



- Desarrollo del programa del validador que realice las respectivas evaluaciones y pruebas para cada equipo, desde las que implican comandos por SSH para verificar configuración del equipo, así como la verificación de los inventariados y estados de tareas activas.
- Proceso de guardado de los resultados del validador en la BBDD de desarrollo.
- Muestreo de resultados del validador para cada equipo en una sección de la página de soporte de PDS en la web.

Fase VI: Pruebas

- Realizar pruebas a pequeña escala, comprobando los diferentes resultados del validador para verificar su correcto funcionamiento.
- Realizar pruebas del proceso de inventariado automático de *routers*.
- Realizar pruebas de la carga correcta de los datos introducidos en la Web, así como de las restricciones planteadas.
- Probar posibles casos específicos de equipos que puedan presentar mayor dificultad.

Fase VII: Elaboración del PFC

- Se recopilará toda la información obtenida anteriormente y se documentará para elaborar el proyecto a entregar.

CAPÍTULO IV

Desarrollo

IV.1.-Fase I: Proceso de investigación y preparación.

IV.1.1. Entrega de requerimientos

Antes que nada, lo primero que se hizo fue una reunión de recolección de requerimientos donde se planteó la necesidad del desarrollo de una herramienta de validación e inventariado de *routers* para aumentar la eficiencia del proyecto, así como también se determinaron las funciones principales y especificaciones que debe cumplir la herramienta.

El proceso completo de validación e inventariado de un *router* empieza desde la finalización de la instalación hecha por el técnico en la sede del cliente. El técnico tiene que subir a la web de PdS los datos correspondientes a la instalación, así como los datos técnicos del *router* y finalmente los datos de la RFS. Al guardar todos estos datos en la web automáticamente quedan registrados en la BBDD del SQL server correspondiente a la web de PdS. Posteriormente el técnico genera el Excel de plantilla de RFS, lo envía al JP del proyecto y sube a *confluence* mediante dos macros programadas en el Excel. Un procedimiento guardado en la BBDD de la web exporta automáticamente los datos correspondientes a las instalaciones nuevas hacia una tabla de validaciones en la BBDD de desarrollos, desde donde se disparan diariamente dos procesos automáticos: uno de inventariado de equipos instalados que registra en una tabla un registro por cada acceso de cada *router* instalado con un formato específico de columnas; el otro proceso es el validador, que pasa por una serie de evaluaciones aquellos equipos instalados que no hayan pasado nunca por el validador, así como

aquellos que sí han pasado pero no están aún validados por completo, por lo que no se han podido facturar. Los resultados del validador se registran en la tabla de validaciones y son mostrados en la web de *PdS* para tener un control sobre las actividades a realizar. Aquellos equipos que no aprobaron todas las validaciones vuelven a pasar por el validador en la siguiente iteración, mientras que los aprobados pasarán a la siguiente factura. La siguiente figura resume el proceso:

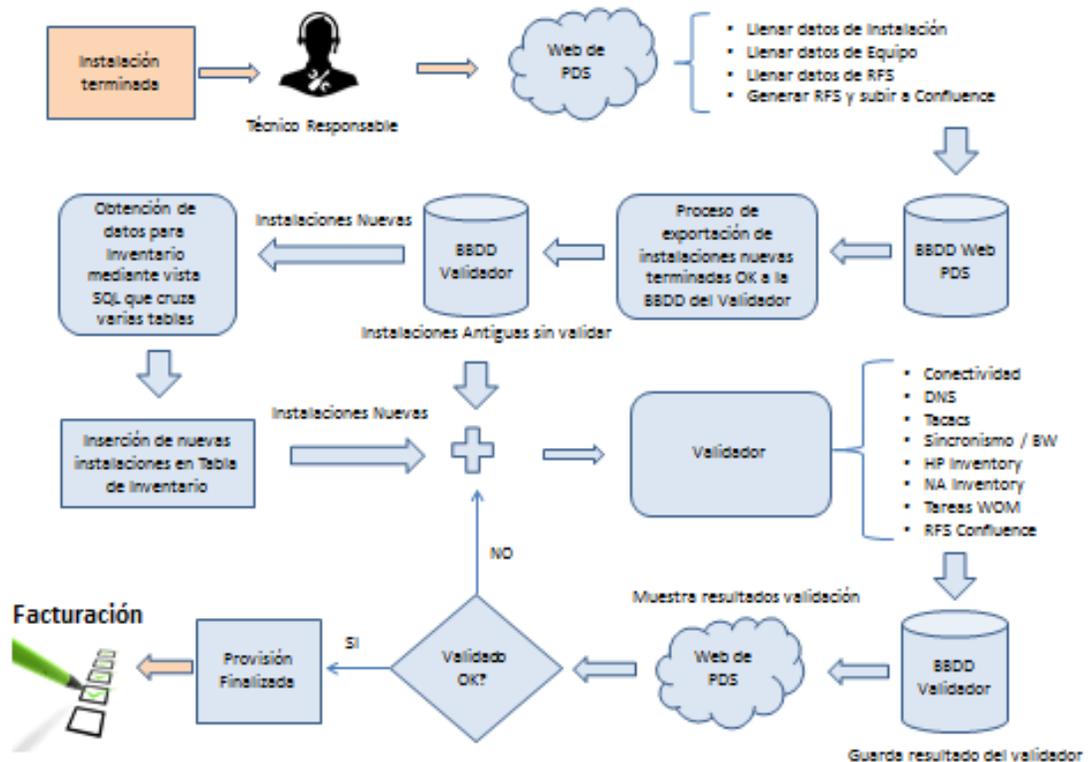


Figura 11: Proceso general del proyecto.

Los requerimientos técnicos específicos recibidos al inicio del proyecto fueron los siguientes:

- Debe comprobar si el *router* está accesible durante el momento de la validación, verificando su conectividad a la red.



- Debe comprobar que se ha hecho la configuración DNS del *router*, asociándolo al *hostname* correspondiente.
- Debe comprobar que el router tenga autenticación Tacacs.
- Debe poder comprobar las velocidades de sincronismo y ancho de banda con las que se configuró el *router*.
- Debe determinar si el *router* si se ha dado de alta en los inventarios webs de HP y NA.
- Debe informar si el *router* tiene tareas en el sistema WOM sin finalizar que impidan la facturación del equipo.
- Debe comprobar que se ha subido una plantilla RFS a *confluence* correspondiente la instalación del *router*.
- Debe guardar un inventario en el que se registren los *routers* instalados en un formato de datos y columnas específico.
- Debe funcionar automáticamente una vez por día con la posibilidad de ejecutarse manualmente mediante un ejecutable en el servidor.
- Debe pasar por las validaciones aquellos *routers* correspondientes a nuevas instalaciones, además de los *routers* antiguos cuyas validaciones no han sido completadas, por lo que no han podido facturarse.
- El inventario debe contener un registro para cada acceso configurado de cada *router* instalado.
- Las datos correspondientes a las validaciones y al inventario deben estar en dos tablas de la BBDD de desarrollos en el Sql server del servidor.
- En la tabla de inventariado se debe guardar fecha de inventariado de cada equipo.
- En la tabla de validaciones deben actualizarse campos de “fecha última validación” y “número de iteraciones” cada vez que se pase por el validador.
- En la web de *PdS* deben introducirse todos los datos correspondientes a la instalación, equipos instalados y RFS. Dichos datos deben guardarse en tablas de la BBDD de la Web.



- Desde la Web de *PdS* se debe poder de generar la plantilla Excel de RFS que se subirá a *confluence*.
- Los resultados del validador deben salir en la Web de *PdS*.

IV.1.2. Lógica de los procesos de validaciones

Una vez obtenidos los requerimientos de validaciones, se procedió a definir los procesos para realizar cada validación:

- Conectividad a la red: para determinar si un *router* en un momento específico está accesible o no, se hará un “telnet” a la *ip loopback* del router por comando SSH. En caso de que el *router* responda, pidiendo autenticación, se validará la conectividad del equipo a la red.
- DNS: para validar que un *router* tiene configurado su DNS con el *hostname* correcto, se realiza un “telnet” al *hostname* del *router* por comando SSH.
- Tacacs: para comprobar que tiene autenticación en Tacacs, se realiza log in en cada *router* con un nombre de usuario y contraseña de Tacacs definidos “*superadmin*” de manera que todo equipo dado de alta en Tacacs debe permitir acceso con dichos datos.
- Velocidad de sincronismo: Para obtener la velocidad de sincronismo de un *router* se ejecutará, una vez dentro del *router*, un “*device*” por comando SSH a la interfaz configurada del *router*, mostrando todos los datos estadísticos del acceso del equipo, y sacando de dicho mensaje las velocidades de transmisión y recepción.
- Ancho de banda: este dato va relacionado al PE (*provideredge*) *router* del equipo, se hace “telnet”, log in y “*enable*” de igual manera pero entrando al PE *router*, dentro se hace un “*show configuration interface*” a la interfaz del PE y saldrá un mensaje con varios datos entre los que se encuentra el ancho de banda.
- Inventario NA: se tiene que comprobar que el *hostname* del *router* esté en el listado del inventario dentro de la web de NA. Se realizará un programa



automático de carga que entre a la web de NA, exporte el inventario a *excel*, y posteriormente se importe dentro de la tabla correspondiente en la BBDD del validador. Para que al momento de validar cada *router*, pueda cruzarse con dicha tabla y validar si se encuentra o no.

- Inventario HP: Misma lógica que con el inventario de NA, la diferencia es que el programa que cargará la tabla en el SQL server no entra a la Web de HP *Inventory*, sino que obtiene el inventario por medio de una extracción Excel enviada al buzón de correo del servidor.
- Tareas de WOM: Cuando se procede a validar cada *router*, se obtienen las tareas activas en WOM por medio de un cruce de tablas, y se valida que ninguna de las tareas pendientes por finalizar impida la facturación, esto se determina mediante una tabla de mapeo que especifica si una tarea es impedimento para facturar o no.
- RFS en *confluence*: Ya existe dentro del proyecto un proceso de carga automática de los nombres de todas las plantillas RFS que se han subido a *confluence* en una tabla de la BBDD de la Web. Para hacer esta validación, sólo se necesita cruzar con esta tabla.

La siguiente figura muestra resumidamente la forma de obtener cada validación:

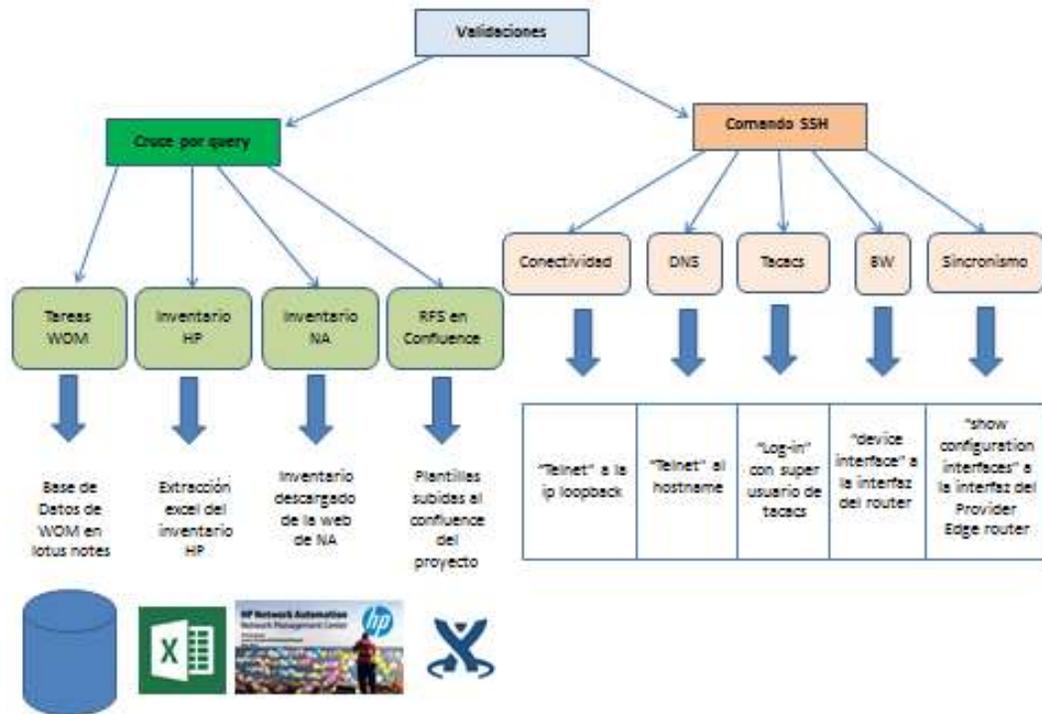


Figura 12: Método de las evaluaciones del validador.

IV.1.3. Lenguajes de programación y herramientas a utilizar

Para el desarrollo de esta herramienta se utilizan diferentes sistemas y lenguajes de programación. Por lo que se requiere tener planteado desde el inicio del proyecto qué tecnologías se usarán y saber en cuáles habrá que fortalecer y aumentar los conocimientos:

- VB.net para el programa principal del validador e inventariado de routers, así como para los procesos de carga de wom, gescir, inventario HP y NA.
- ASPx, HTML, CSS y JavaScript para la Web de PdS.
- Selenium como framework en el programa de carga del inventario de NA para facilitar la navegación en la web de NA.
- SSH para realizar validaciones técnicas sobre los routers.
- SQL Server, BBDD de desarrollo y de la Web. Tablas, vistas y procedimientos.



- Access para la BBDD de GESCIR.
- Lotus Notes para la BBDD de WOM.
- Excel para la plantilla RFS

IV.1.4 Valoración temporal

Inicialmente se estimó para el desarrollo de la herramienta un tiempo de 6 semanas, incluyendo pruebas y documentación del diseño técnico. No se toma en cuenta el tiempo de realización del presente trabajo. En esta valoración se toman 8 horas efectivas por día de trabajo. Distribuido de la siguiente manera:

Tarea	Tiempo estimado (horas)
Planteamiento y preparación	16
Cambios Web PdS	32
Creación de tablas, vistas y procedimientos en SQL Server	24
Modificación de programas de carga de Wom y Gescir	8
Programa de carga NA	16
Programa de carga HP	12
Desarrollo programa validador e inventariado	80
Preparación de ejecutable y tarea automática en el servidor	4
Pruebas	32
Documentación (Diseño técnico)	16

Tabla 2: Valoración estimada de tiempo.

IV.2.-Fase II: Página de control, gestión y soporte de PDS dentro de la Web

Dado que los datos de entrada del proyecto se introducirán desde la Web, se procedió a hacer una página dentro de la web de *PdS*, llamada “Soporte Datos”. A continuación se muestra el menú inicial de la web de Provisión:

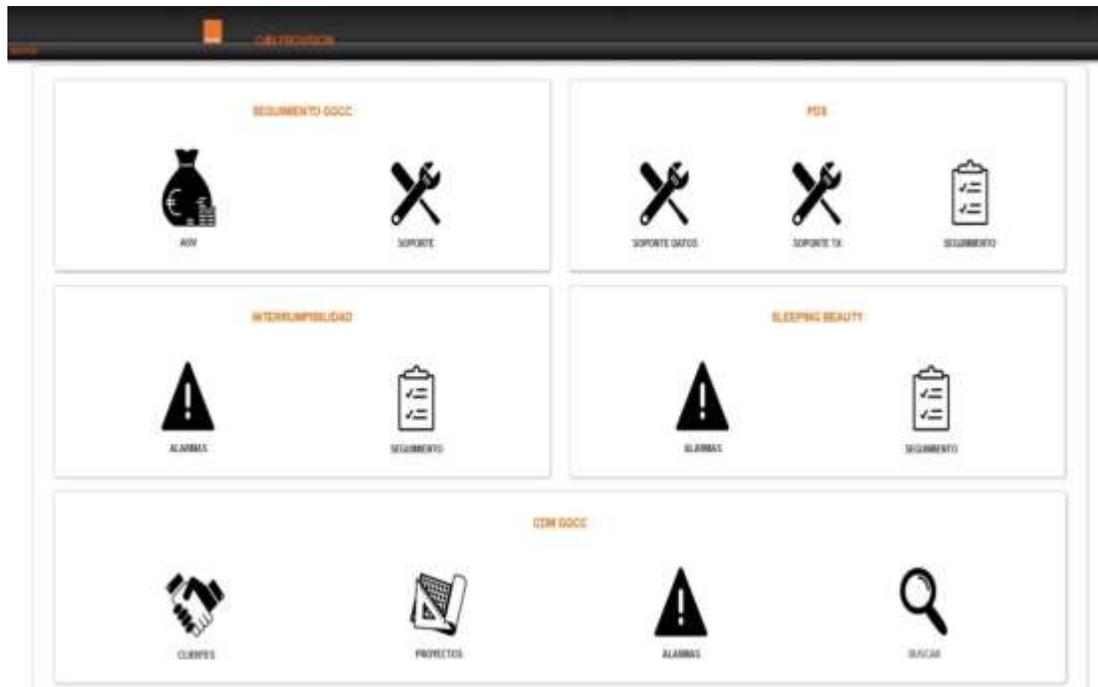


Figura 13: Menú inicial Web Provisión.

IV.2.1 Estructura de la página

Se acordaron los siguientes cuadros de datos o partes dentro de la página:

IV.2.1.1 Información WOM

Parte inicial de la página que contiene un cuadro de texto para introducir el código WOM de un proyecto, al presionar “Enter” o darle al buscador se crea una conexión a la BBDD de la Web y se extraen los datos relacionados al proyecto para ser mostrados en la página. Dentro de esta parte también se encuentran datos generales administrativos del proyecto como lo son: Cliente, JP, Ítem, acción, etc., estos datos

son sólo etiquetas que se muestran, no pueden modificarse ni introducirse desde la web. Estos datos se encuentran almacenados en la tabla GGCC_PROYECTO.

IV.2.1.2 Estado

Contiene un esquema de los estados secuenciales, con excepción de “Incidencia” que es un estado no secuencial, y al buscar un proyecto específico se trae la información de su estado desde la BBDD y se pinta de color naranja la caja del estado correspondiente. Al igual que la parte anterior, los datos que se muestran en esta parte son etiquetas que no se pueden modificar desde la web. Estos datos son: Fecha Incidencia, Causa Incidencia, Estado y Fecha Próxima Acción. Estos datos se encuentran almacenados en la tablas de GGCC_PROYECTO_ESTADO y GGCC_PROYECTO_INCIDENCIAS, mostrándose en la web de la siguiente forma:

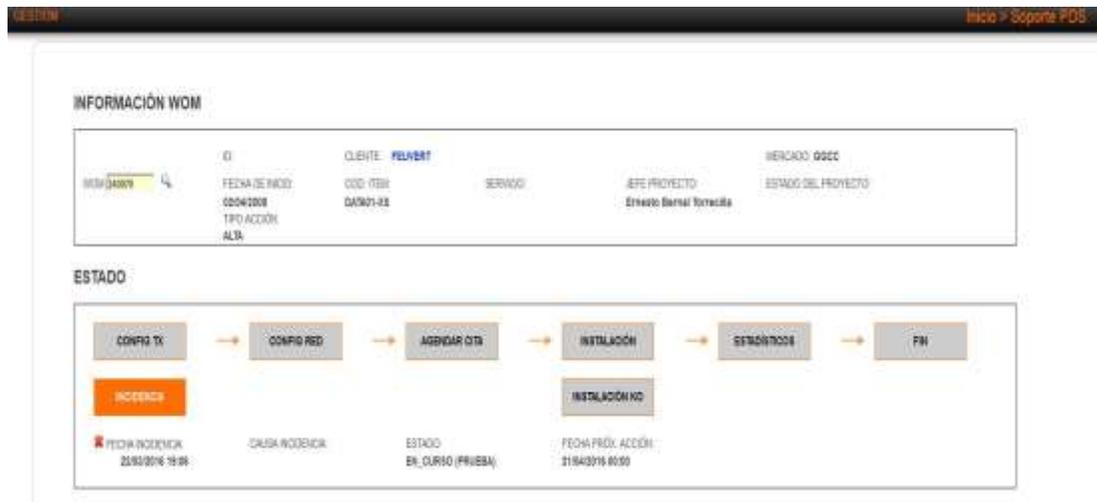


Figura 14: Datos relacionados al Proyecto.

IV.2.1.3 Instalación- Datos

Aquí se encuentran los datos de la instalación que tiene que realizar el técnico una vez finalizada ésta. Estos datos son: Fechas de inicio y fin de instalación, estado de la instalación, técnico responsable, motivo KO en caso de que no se haya podido completar la instalación, número de incidencia, tipo de actuación y observaciones.

Una vez el técnico haya completado los datos de la instalación se da al botón registrar, almacenándose los datos en la BBDD de la Web, y actualizando el *grid* que muestra la tabla de todas las visitas de instalaciones que se han hecho al cliente para el WOM en cuestión, como se puede ver en la siguiente figura:

Figura 15: Datos de la instalación.

Los datos correspondientes a esta parte de almacenan en la tabla de PDS_ACTIVIDAD_INSTALACIONES_VISITAS, cuyo diseño se muestra a continuación:

Columna	Tipo de Dato
ID	bigint
WOM_PROYECTO	bigint
NUM_VISITA	int
FECHA_INICIO	datetime
FECHA_FIN	datetime
ESTADO	nvarchar(50)
MOTIVO_KO	nvarchar(255)
NUM_INCIDENCIA	nvarchar(255)
COMENTARIO	nvarchar(MAX)
RESP_INSTALACION	nvarchar(255)
TIPO_ACTUACION	nvarchar(255)
SUBTIPO_ACTUACION	nvarchar(255)

Tabla 3: Diseño tabla visitas.



Columna	Tipo de dato
WOM_PROYECTO	bigint
ID_ACCESO	int
RED	nvarchar(200)
TECNOLOGIA	nvarchar(200)
MODALIDAD	nvarchar(200)
ANCHO_BANDA	nvarchar(200)
NUMERO_ADMINISTRATIVO	nvarchar(200)
CODIGO_ACCESO	nvarchar(200)

Tabla 4: Diseño tabla de accesos.

Columna	Tipo de dato
ID	int
WOM_PROYECTO	bigint
CODIGO_LINEA	nvarchar(255)
HOSTNAME	nvarchar(255)
ORDEN_NEMONICO	nvarchar(255)
TIPO_ROUTER	nvarchar(255)
ROUTER_CONCENTRADOR	nvarchar(255)
INTERFAZ_CONCENTRADOR	nvarchar(255)
IP_LOOPBACK	nvarchar(255)
SUBRED_CONEXION	nvarchar(255)
WAN_RC	nvarchar(255)
WAN_CPE	nvarchar(255)
RD	nvarchar(255)
VRF	nvarchar(255)
IOS	nvarchar(255)
QoS	nvarchar(255)
SERIAL_ROUTER	nvarchar(255)
DIRECC_PRIVADO	nvarchar(255)
DIRECC_PUBLICAS	nvarchar(255)
AAC_USUARIO	nvarchar(255)
AAC_DOMINIO	nvarchar(255)

Tabla 5: Diseño tabla de CPE.

IV.2.1.5 Instalación- RFS

En esta parte el técnico debe introducir datos para la plantilla de RFS que debe generar, subir a *confluence* y enviar al JP del proyecto vía email. La opción de generar la plantilla RFS está bloqueada hasta que no se haya dado a Guardar en los datos introducidos. Una vez guardado los datos, se almacenan en la tabla PDS_ACTIVIDAD_RFS. Al generarse la plantilla, el técnico puede, por medio de unas macros programadas, subir la plantilla a *confluence* y mandar el correo al jefe del proyecto. El cuadro de datos de RFS se muestra en la web de la siguiente manera:

Figura 17: Datos de la RFS.

A continuación se muestra el diseño de la tabla de RFS:

Columna	Tipo de dato
ID_RFS	int
WOM_PROYECTO	bigint
ID_CPEs	int
COD_CPE	nvarchar(255)
N_ACCESOS	int
TIPO_SERVICIO	nvarchar(255)
TECNOLOGIA	nvarchar(255)
ROUTER	nvarchar(255)
PE	nvarchar(255)

Tabla 6: Diseño tabla de RFS.

IV.2.1.6 Comentarios y Archivos

Al final de la página se colocó una parte para que el técnico pueda registrar comentarios y subir adjuntos que estén relacionados a la instalación. Como se observa a continuación:

The image shows two side-by-side screenshots of a web application interface. The left screenshot is titled 'COMENTARIOS' and shows a form with a text area for comments, a 'Añadir' button, and a 'Cambiar' button. The right screenshot is titled 'ARCHIVOS' and shows a table with columns for 'ID', 'DESCRIPCIÓN', and 'ACCIONES'. Below the table is a search field and a 'Descargar los archivos' button.

ID	DESCRIPCIÓN	ACCIONES
1	Informe de actividad primer día de...	[Iconos]
2	Informe de actividad segundo día de...	[Iconos]
3	Informe de actividad tercer día de...	[Iconos]

Figura 18: Comentarios y Archivos.

Los datos se almacenan en las tablas PDS_ACTIVIDAD_ADJUNTOS y PDS_ACTIVIDAD_COMENTARIOS.

IV.2.1.7 Gestión

En esta parte el técnico introduce los datos relacionados a la gestión del equipo posterior a su instalación. La parte de gestión también incluye al final un *grid* que contiene los resultados del validador. El *grid* se carga con los datos de la tabla de validaciones y estos datos no pueden modificarse desde la Web. La siguiente figura muestra un ejemplo de cómo se mostrarían los resultados del validador en la web:

GESTIÓN

FIN GESTIÓN: 24/08/2016 05:37

HP TACACS Tk Ocean 2131231231 NA Jefe de Proyecto: Antonio Serra Moreno

[Ir a HP Inventory](#)

RES	CPE	ULTIMA VALIDACIÓN	ITERACION VALIDACIÓN	ACCESO IP	DNS	TACACS	HP	NA	TAREAS WOM	RESULTADO	TRATAR RES	CORREO A JP
922	rcbmm83	25/08/2016 14:43	2	✓	✓	✓	✓	✓	✓	✓	✗	✓
922	rcbmm83	25/08/2016 14:43	2	✓	✓	✓	✓	✓	✓	✓	✗	✓
923	rpaimr988	28/08/2016 2:06	5	✓	✓	✓	✓	✓	✓	✓	✗	✓

[Guardar](#) [Cancelar](#)

Figura 19: Tabla de Gestión.

IV.2.2 Cambios a nivel de código

Para agregar la página de soporte datos a la web de *PdS* se tuvo que modificar el código de la web. Se añadió la nueva página al *SiteMap* de la web, y se programó para que se acceda a ella desde el menú inicial.

Para la página se tuvo que desarrollar el *SoportePDS.aspx* que contiene el código HTML de diseño de la página, para esta parte fue necesario acordar previamente cuáles campos correspondían a cuadros de texto, etiquetas, *radio-buttons*, botones, *grids*, links, etc. También se hizo el código del *SoportePDS.aspx.vb* que contiene la lógica, condiciones, procedimientos, funciones, validaciones y conexiones a la base de datos para ir realizando las *queries* correspondientes a las diferentes acciones.

También se agregaron algunos nuevos estilos para la página en las clases CSS y algunas restricciones y comprobaciones de campos por medio de funciones en *JavaScript*.

En el **Anexo A** se puede ver parte del código añadido a la web de PdS.



IV.3.-Fase III: Modelado de Datos de entrada al proyecto

Una vez definidas las tablas de la BBDD de la Web que contienen los datos necesarios de entrada al validador, se procedió a la creación de las tablas de validaciones e inventariado en la BBDD de desarrollos. Además se desarrolló un proceso de exportación de datos de la BBDD de la Web hacia la de desarrollos y se agregaron los campos necesarios para inventariar a otros procesos de carga.

IV.3.1. Tabla del validador

La tabla principal desde la cual se ejecuta el validador debe contener datos administrativos y generales del proyecto, así como los datos ingresados en la web de *PdS* y además datos relacionados con los resultados del validador. En esta tabla debe insertarse un registro por cada acceso de cada *router* instalado, debido a que diferentes accesos tienen diferentes valores para algunas columnas. Básicamente los datos de la tabla pueden clasificarse en los siguientes tipos:

Nivel de instalación: son aquellas columnas que se cargan con datos provenientes de las tablas de instalaciones, visitas y RFS de la Web de *PdS*. Estos datos son comunes para todos los registros en la tabla del validador relacionados a la misma instalación.

Nivel de equipo: son aquellas columnas que se cargan con datos provenientes de la tabla de *CPEs* de la Web de *PdS*. Estos datos son comunes para todos los registros en la tabla del validador relacionados al mismo *router*.

Nivel de acceso: son aquellas columnas que se cargan con datos provenientes de la tabla de accesos de la Web de *PdS*. Estos datos son únicos por registro.

Nivel de registro: Son datos únicos por registro independientes de los otros registros en la tabla. Corresponden al ID, el cual es la *Primary Key* de la tabla que se asigna automáticamente al insertarse el registro, no puede haber más de un registro en toda la tabla con un determinado ID; y los campos que se rellenan en el validador, los



cuales pueden ser de control (aquellos campos relacionados al paso por el validador pero no al resultado específico de una evaluación) o de evaluación (los que corresponden a una evaluación concreta).

La tabla de validaciones se modifica sólo por dos procesos: el primero es el procedimiento SQL automático que inserta los datos correspondientes a nuevas instalaciones, que no se encuentran aún registradas en la tabla de validaciones, desde las tablas de la BBDD de la web de *PdS*; el segundo es el programa del validador, el cual actualiza los campos correspondientes a los resultados y control de las validaciones.

El nombre en la BBDD de la tabla de validaciones es `IS2_IMPORTED_DATA_DEF`. En el **Anexo B** se puede observar el diseño completo de la tabla.

IV.3.2. Tabla de inventario

La tabla de inventariado debe seguir un formato de columnas específicos definidos por el cliente. Anteriormente al desarrollo de esta herramienta, el inventario de *routers* instalados era un Excel que se llenaba manualmente y estaba compartido en red. Esto se modificó para que sea una tabla en la BBDD de desarrollos.

Dado que no todos los campos requeridos para el inventario de un equipo se pueden obtener desde la tabla de validaciones, se requiere cruzar con las tablas de *WON* y *GESCIR* para poder inventariar correctamente un equipo.

Esta tabla sólo se modifica cuando se agregan registros nuevos correspondientes a las instalaciones del día, durante la ejecución del programa de validador e inventariado.



El nombre en la BBDD de la tabla de inventariado es INVENTARIO_CPES_DEF. En el **Anexo C** se muestran los campos del Inventario y su respectiva tabla de origen.

IV.3.3. Procedimiento de exportación de datos entre las BBDD

Dado que el programa del validador e inventariado se ejecuta sobre los *router* de la tabla de validaciones, es necesario que un proceso automático cargue diariamente los datos correspondientes a las nuevas instalaciones desde las tablas de la BBDD de la Web hacia la tabla de validaciones. Este proceso se realiza en dos partes.

IV.3.3.1 Carga_PDS_Web_SQL

Es una vista SQL que cruza las diferentes tablas de la BBDD de la Web de PdS relacionadas a la página de soporte datos, filtrando los datos relacionados a instalaciones nuevas del último día, obteniendo aquellos campos necesarios para el validador y que pueden ser insertados en algunas columnas de la tabla de validaciones.

En el **Anexo D** se puede ver el diseño completo de la vista.

IV.3.3.2 Insert_Into_IS2

Es un procedimiento SQL que inserta el resultado de la vista Carga_PDS_Web_SQL dentro de la tabla de validaciones. Se ejecuta una vez por día de manera automática desde el programa del validador aunque puede ejecutarse manualmente también desde el SQL Server Management Studio.

En el **Anexo E** se puede ver el diseño completo del procedimiento.

IV.3.4. Modificación tabla y proceso de carga WOM

Debido a que el Orange solicita que el inventario de *routers* instalados cuente con un determinado formato de columnas, las cuales no corresponden completamente a datos que se puedan obtener de la Web de *PdS*, es necesario que aquellos datos que sean requeridos para el inventariado de un equipo se obtengan de los sistemas correspondientes.

Entre los campos requeridos para inventariar que no se obtienen de la Web, hay 3 que se pueden obtener desde la BBDD de WOM, los cuales son:

- Dirección Sede
- Código del cliente
- Servicio

Desde la BBDD de WOM ya existía un proceso de carga automático de datos hacia la tabla WOM_IMPORTED_DATA de la BBDD de desarrollos. Este proceso de carga está desarrollado en Vb.net y las únicas modificaciones que tuvieron que hacerse fueron:

- Añadir los 3 campos a la consulta de exportación de la BBDD de WOM.
- Añadir los 3 campos al modelo del programa de carga a la BBDD de desarrollos.

IV.3.5. Modificación tabla y proceso de carga GESCIR

Entre los campos requeridos para inventariar que no se obtienen de la Web, hay 2 que se pueden obtener desde la BBDD de GESCIR, los cuales son:

- Nombre contacto cliente
- Teléfono contacto cliente

Al igual que con WOM, desde la BBDD de GESCIR ya existía un proceso de carga automático de datos hacia la tabla GESCIR_IMPORTED_DATA de la BBDD de desarrollos. Este proceso de carga está desarrollado en Vb.net y las modificaciones que tuvieron que hacerse fueron las mismas que para el caso anterior.

IV.4.-Fase IV: Procesos de carga automáticos de los Inventarios

IV.4.1. Tabla y proceso de carga NA

El proceso de obtención del inventario de NA es el siguiente:

1.- Ir a la Web de HP Network *Automation*, se requiere que estés conectado a la VPN de Orange y con el proxy configurado. Hacer Log-in introduciendo usuario y contraseña en la página que se muestra a continuación:



Figura 20: Web de NA.

2.- Seleccionar en el menú principal “*Devices*”

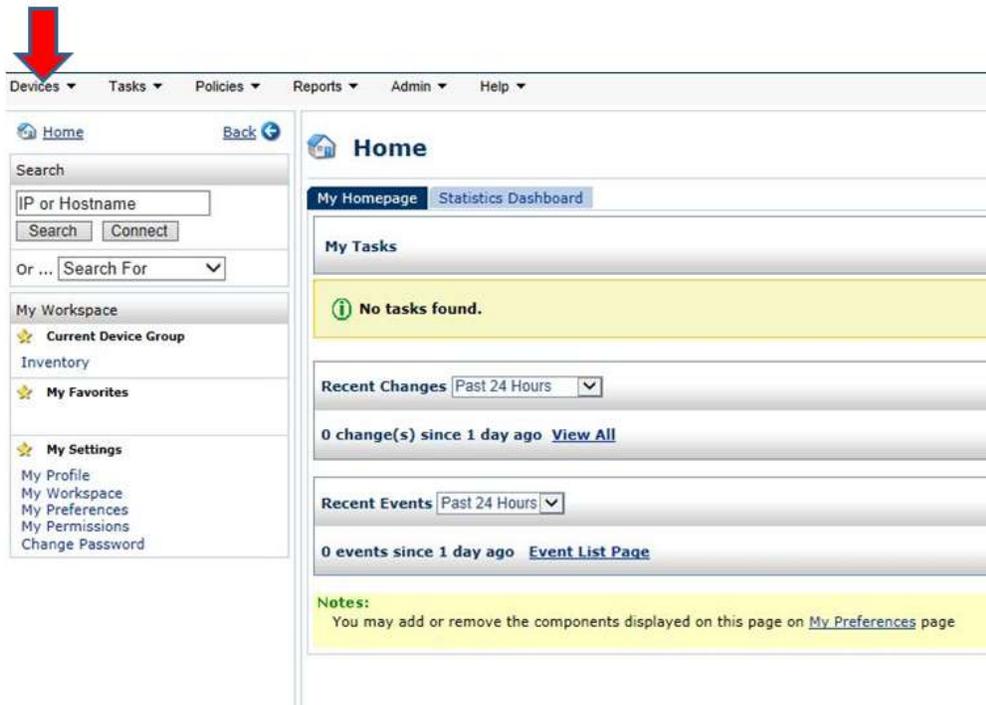


Figura 21: Selección *Devices*.

3.- Seleccionar “*Inventory*”



Figura 22: Selección *Inventory*.

4.- Una vez se muestre todo el inventario hay que seleccionar “Exportar a Excel”.

Se desarrolló un programa en vb.net que realiza el proceso de obtención del inventario y lo carga en la tabla NA_IMPORTED_DATA de la BBDD de desarrollos. El programa utiliza *selenium* como *framework* que facilita la



navegación web a nivel de código. En el **Anexo F** se puede ver el código del programa de carga de NA.

Dado que al validador sólo le interesa comprobar que un *router* se encuentra inventariado, en el proceso de carga no se importan todos los datos hacia la tabla de la BBDD. Los únicos datos que se importan son:

- *Hostname* del *router*
- *IP Loopback* del *router*
- Modelo del *router*

IV.4.2. Tabla y proceso de carga HP

A diferencia de la web de NA, la web de HP *Inventory* no permite la obtención de todo el inventario de *routers*. Dado que esta parte del proceso no se puede automatizar por completo, la solución a la que se llegó con el cliente fue que diariamente se enviaría una extracción del inventario de HP con sólo los datos de *Hostname* e *IP Loopback* de los *router* en el inventario en un documento Excel al buzón de correo del servidor, desde el cual se hizo un proceso simple de carga que vacía mediante un *truncate* la tabla de HP_IMPORTED_DATA en la BBDD de desarrollos, y posteriormente importa los datos del Excel dentro de la tabla.

IV.5.- Fase V: Desarrollo del validador e inventariado

En las fases anteriores se explicó el modelado y procesos de tratado de los datos en la Web, BBDD de Web y desarrollos con sus tablas y origen de datos. La figura 23 muestra el proceso por el que pasan los datos de una instalación durante toda la herramienta:

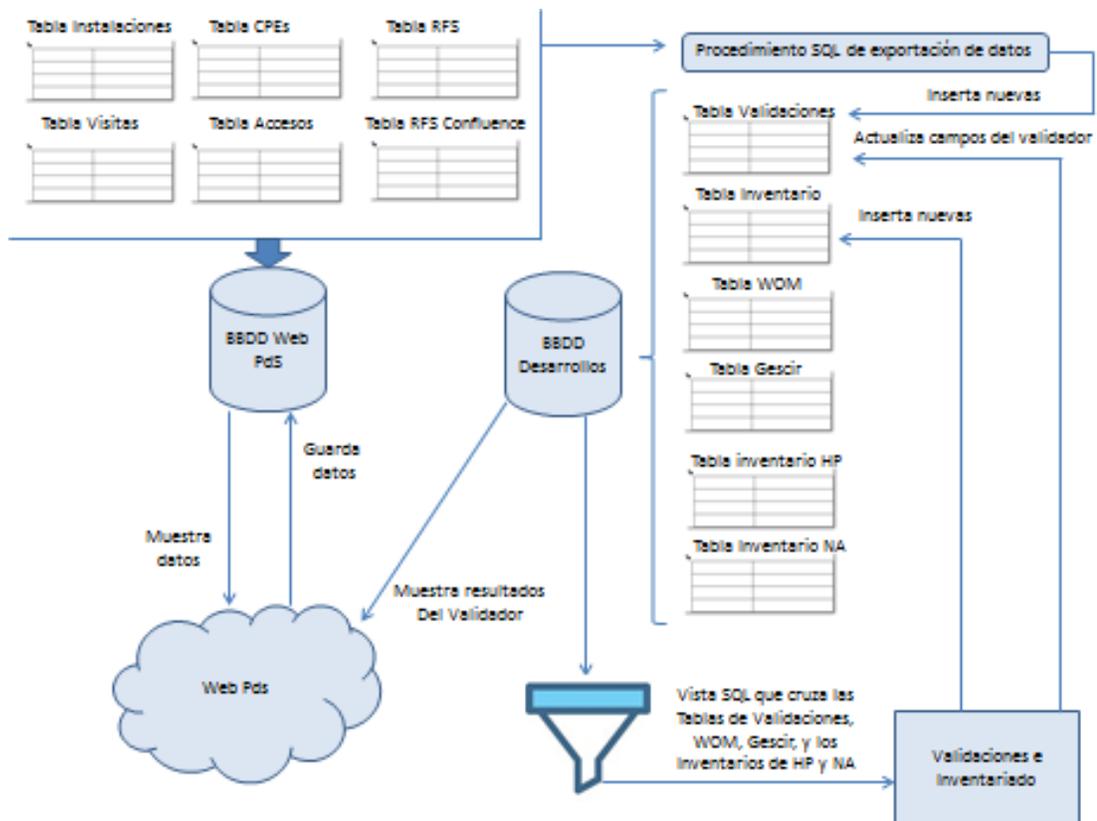


Figura 23: Procesos de los datos de una instalación.

En esta fase se explicará el programa encargado de hacer finalmente la validación e inventariado de los datos de los *routers* instalados para ser facturados. Se desarrolló un programa en Vb.net para validar e inventariar automáticamente los *routers* instalados. En esta fase se presenta como objetivo conseguir que aquellos *routers* instalados que no han sido inventariados se registren en la tabla de inventario, así como obtener los resultados de las validaciones y actualizar los campos correspondientes al validador en la tabla de validaciones.

Los campos que el validador actualiza sobre la tabla de validaciones se pueden clasificar en 2 tipos:



1.- Campos correspondientes a resultados de validaciones: Son aquellos campos que se son de tipo booleano y van relacionados a una validación específica: Conectividad, DNS, *Tacacs*, Tareas WOM pendientes, etc.

2.- Campos correspondientes al control de las validaciones: son campos que se actualizan por cada paso del *router* por el validador, y están relacionados con el validador global y no con una validación específica. Estos son:

- Fecha última validación: se actualiza a fecha actual con cada iteración
- Número de iteraciones: Entero que cuando se inserta el registro en la tabla tiene valor por defecto = 0, y se aumenta una unidad con cada iteración.
- Resultado Final: Por defecto es 0, se pasa a 1 cuando los resultados de todas las evaluaciones específicas son 1.
- Tratar RFS: Por defecto es 1, se pasa a 0 cuando el Resultado final se cambia a 1, o se puede cambiar manualmente cuando por alguna razón deja de ser necesario que se evalúe el *router*, ya sea que se haya cancelado el proyecto, o haya entrado en parada de reloj, o se haya pasado a facturación independientemente de que el resultado final sea 0 en casos excepcionales.

La siguiente figura muestra las principales funciones que realiza el programa Vb.net de validaciones e inventariado:



Figura 24: Partes del programa de validación e inventariado.

Se describirán las principales funciones y procesos de cada parte del programa para una explicación más ordenada. Gran parte del código del programa se puede ver en el **Anexo G**.

IV.5.1 Tarea de ejecución y log

Esta parte corresponde a la tarea de Windows programada para ejecutarse desde el servidor cada día. La tarea inicia un *.bat* que simplemente ejecuta el archivo ejecutable del programa del validador e inventariado. Deja un archivo *.txt* de log para saber enfocar errores en caso de que algún día el programa se ejecute incorrectamente o falle. Durante todas las partes del código se van dejando trazas en el log que especifican cómo va el funcionamiento del programa.

A continuación se muestra un trazo del código que muestra la creación del archivo de log:

```
Try
    mObjLog = New Logging.Log("Validador Técnico")
    Dim logsFolder AsString = My.Application.Info.DirectoryPath +
        "\LOGS\ValidadorTecnico - "& Now().ToString("dd-MM-yyyy-HH_mm") & ".log"

    mObjLog.DefaultFileLogWriter.Location = Logging.LogFileLocation.Custom
    mObjLog.DefaultFileLogWriter.CustomLocation = logsFolder

    mObjLog.DefaultFileLogWriter.Location =
        Logging.LogFileLocation.ExecutableDirectory

    Catch ex AsException
        Console.WriteLine(ex.ToString)

    Finally

        EndTry
```

IV.5.2Ejecución procedimiento

Es la primera parte del programa, se hace una conexión a la BBDD de desarrollos y se ejecuta el procedimiento INSERT_INT0_IS2 que carga los datos de nuevas instalaciones desde las tablas de la BBDD de la Web en la tabla de validaciones.

```
objSqlConnection = New SQLServerDataConnection("user", "password",
    "bdd", "ip")

Dim query1 As String = "EXECUTE INSERT_INT0_IS2 "
objSqlConnection.ExecuteNonQuery(query1)
```

IV.5.3Evaluaciones técnicas por comando SSH

Filtra registros con algún fallo de validación técnica y pasa por la función de validaciones por comandos SSH.

Primero se ejecuta una *query* que filtra de la tabla de validaciones aquellos registros que tengan: Código RFS definido; Resultado Final = 0; Tratar RFS = 1; Al menos una validación técnica por comando SSH = 0.

Este filtro sólo obtiene aquellos *routers* que tengan que pasar por las validaciones técnicas que se realizan por comandos SSH. El resultado de la *query* se guarda en una tabla de datos:

```
Dim cpe_data As DataTable = Nothing

    Try
        mObjLog = New Logging.Log("Router connectivity checking")

        cpe_data = objSqlConnection.ExecuteQuery("SELECT * FROM [BBDD].[dbo].[IS2_IMPORTED_DATA_DEF] WHERE [Código RFS] IS NOT NULL AND [Resultado Final] = 0 AND [Tratar RFS]=1 AND ([Acceso IP]=0 OR [Configuración DNS]=0 OR [Alta TACACS]=0)")
```

Posteriormente se pasa cada registro de la tabla de datos por las validaciones mediante una serie de comandos SSH realizados al *router*. El código relacionado a estas validaciones se puede ver en el Anexo X.

Para crear un comando se introducen aparte de los parámetros que se vayan a introducir, las posibles respuestas esperadas. Por ejemplo:

```
Dim objTelnetConnectionCommand As Command = New Command("/usr/bin/telnet", "", "", "username:;user:;password:", "Error while connecting through telnet")
```

En este caso, se define que para el caso de que el comando de “telnet” se ejecute correctamente, las posibles respuestas corresponderían a la petición de autenticación:

- *username:*
- *user:*
- *password:*

Mientras que en caso de que el comando no se ejecute correctamente la respuesta de error se define que sería: *Error while connectingthrough telnet*.

Los dos campos vacíos corresponden a los parámetros del comando, por lo que antes de hacer la ejecución del comando se debe definir al menos uno de ellos, ya sea el parámetro de IP o *hostname* al cuál hacer el telnet, ya que de lo contrario el comando no se podrá ejecutar.

La ejecución de un comando SSH se hace mediante la clase *Check Router Analyzer*, que tiene como parámetros una serie de comandos SSH a ejecutar.

Si los comandos correspondientes a las validaciones de conectividad, DNS y Tacacs se ejecutan correctamente, se guarda un 1 en las respectivas variables de resultados de cada evaluación. Para los comandos que devuelven mensajes de texto de velocidades de sincronismo y ancho de banda, se procede a extraer de dichos mensajes los valores específicos y se guardan en variables de texto.

IV.5-4 Vista SQL de cruce

Una vez terminados de ejecutarse las evaluaciones de comandos SSH, se ejecuta una vista SQL llamada *CERTIFICADOR_IS2_FINAL_DEF*, cruzando las tablas de Validaciones, con las de:

- WOM: el cruce con WOM permite obtener 3 de los datos necesarios para inventariar. Además se saca la validación de si quedan tareas pendientes sin finalizar que impidan la facturación del *router* en la BBDD de WOM.
- GESCIR: permite obtener dos datos necesarios para inventariar un registro.
- HP: permite validar si se ha registrado al *router* en el inventario de HP.
- NA: permite validar si se ha registrado al *router* en el inventario de NA.

En el **Anexo G** se puede observar el contenido de la vista.

IV.5.5 Inventariado

Con el resultado de la vista guardado en una tabla de datos se procede hacer una consulta para validar si el router se encuentra ya registrado en el inventario, y en caso



que no se haya inventariado, se ejecuta una función que inserta los datos correspondientes al inventario dentro de la tabla INVENTARIO_CPES_DEF.

```
query = "SELECT * FROM
[orange_cdm_industrializado].[dbo].[INVENTARIO_CPES_DEF] WHERE [IS2_ID]=" &
id.ToString

If objSqlConnection.ExecuteQuery(query).Rows.Count = 0 Then
    insertarRouterInventarioCPE(row)
End If
```

IV.5.6 Evaluaciones finales

De la vista se obtienen el resultado de las validaciones restantes, por lo que se procede a obtener dichos valores y guardarlos en las variables de resultados.

```
If row("OK Tareas WOM") = 1 Then
    okTareaswom = True
End If

If row("Alta HP Inventory") = 1 Then
    okHPINV = True
End If

If row("Alta NA Inventory") = 1 Then
    okNA = True
End If
```

IV.5.7 Actualización de campos

Se procede finalmente a actualizar los campos de: fecha última validación con la fecha actual; número de iteraciones sumándole 1; Resultado final según las variables de los resultados de las validaciones; Tratar RFS según el resultado final.

Por último se ejecuta una *query* que actualiza y guarda los campos de control y evaluaciones del validador dentro de la tabla de validaciones.

IV.6.- Fase VI: Pruebas

Para comprobar el correcto funcionamiento de la herramienta, encontrando fallos, incidencias y mejoras, se realizaron diferentes pruebas a los desarrollos que se implementaron. A continuación se resumen a gran escala las pruebas realizadas. En el Capítulo V se plantean los resultados y sus análisis.

IV.6.1 Pruebas de la página de soporte datos PDS

- Comprobar que los datos del proyecto se muestren correctamente en la web al realizar una búsqueda de un WOM.
- Comprobar que los combo box, tablas, comentarios y archivos se carguen correctamente en la web.
- Comprobar el correcto funcionamiento de los botones de guardar de las diferentes partes de la página, verificando que los datos se almacenan en sus respectivas tablas de la BBDD de la manera esperada.
- Probar el funcionamiento del proceso de subir plantilla de configuración del *router* a la web.
- Probar el proceso de generación de la RFS desde la página.
- Comprobar que las condiciones y restricciones de algunos botones se cumpla.
- Comprobar que se muestren correctamente los resultados del validador.

IV.6.2 Pruebas del proceso de exportación

- Comprobar que todos los datos correspondientes a instalaciones del último día se filtren correctamente por la vista del proceso de exportación de datos hacia la BBDD de desarrollos de manera que sean los únicos datos que se exporten y que no falte ninguno.
- Comprobar que no se inserten registros repetidos en la tabla de validaciones.

IV.6.3 Pruebas de los procesos de carga de los inventarios

- Probar la carga del inventario de HP a su correspondiente tabla en la BBDD de desarrollo. Que todos los datos se importen correctamente y que la cantidad de registros en la tabla sea igual a la del inventario.
- Probar la carga del inventario de NA a su correspondiente tabla en la BBDD de desarrollo. Que todos los datos se importen correctamente y que la cantidad de registros en la tabla sea igual a la del inventario.

IV.6.4 Pruebas del proceso de inventariado

- Probar la vista de cruce de tablas de validaciones, WOM y GESCIR de manera que se obtengan correctamente todos los datos requeridos para inventariar.
- Probar el proceso de inventariado, verificando que la cantidad de registros nuevos en el inventario corresponda con la cantidad de registros nuevos en validaciones.

IV.6.5 Pruebas del validador

- Verificar que se filtren correctamente los registros que pasarán por las evaluaciones técnicas de comandos SSH.
- Probar el proceso para validar la conectividad a la red de un *router*.
- Probar el proceso para validar la configuración DNS.
- Probar el proceso para validar la autenticación *Tacacs*.
- Probar el proceso para obtener la velocidad de sincronismo del *router*.
- Probar el proceso para obtener el ancho de banda del PE *router*.
- Probar el proceso para verificar que la plantilla RFS esté subida a *confluence*.
- Probar el proceso para verificar que el proyecto no tenga tareas de WOM sin finalizar que impidan la facturación del equipo.
- Probar el cruce con las tablas de inventarios de HP y NA para comprobar el inventariado del *router*.



- Probar que los campos de control de validaciones se actualicen correctamente al finalizar las evaluaciones.
- Probar que los campos correspondientes a los resultados de validaciones se actualicen correctamente.
- Probar que el programa se ejecuta correctamente una vez al día de manera automática según la tarea programada en el servidor
- Probar que el trazado del log que deja el validador cada vez que se ejecute sea correcto.

IV.7.- Fase VII:Elaboración del PFC

Evidentemente, esta fase es de gran importancia dado que es donde se reflejan todos los aspectos relevantes en la elaboración del Proyecto Final de Carrera. Además, la preparación del tomo forma parte del trabajo, representando así parte de su evaluación y lo más importante su culminación.

CAPÍTULO V

Resultados obtenidos

Los resultados de las pruebas realizadas en diversos escenarios fueron satisfactorios. Surgieron incidencias y pequeños errores a la hora de realizar la instalación de los *routers* que posteriormente fueron corregidos. Todas las comprobaciones mencionadas en la fase 6.5 del capítulo anterior fueron realizadas correctamente.

Sin embargo, hubo dificultades con pruebas de instalaciones que requerían la instalación de más de dos *router* con más de dos accesos a configurar cada uno. Principalmente esto se debió a que inicialmente se planteó en la web un espacio máximo de dos *routers* por instalación, cada uno con un máximo de dos accesos configurables. Estos casos excepcionales no se habían tenido en consideración al momento de los requerimientos.

Se ha obtenido un mayor control sobre los estados de los equipos ya instalados pendientes por facturar, proporcionando las acciones a tomar sobre cada equipo dependiendo de los resultados de sus validaciones.

Se ha logrado sustituir el inventario de *routers* que anteriormente era un Excel por una tabla de carga y modificación automática en la BBDD en el servidor del proyecto.

Los técnicos responsables de las instalaciones, así como el personal de *PdS* encargado del inventariado de equipos, han valorado de manera positiva la herramienta, que facilita y simplifica los esfuerzos necesarios para completar la



gestión de los *routers* instalados para poder ser facturados, optimizando las funciones del área de la provisión del servicio.

La factura del primer mes posterior a la implementación de la herramienta aumentó en más de 10% con respecto a los meses anteriores. Esto se debe a que el proceso de validaciones e inventariado se hizo más eficientemente, consumiendo menos tiempo y recursos. El tiempo medio de validación de los equipos instalados se redujo en un dos días aproximadamente. El tiempo medio de inventariado por *router* pasó de 2 días a 1, todo esto sin la necesidad de gasto temporal por parte del personal del proyecto.

CAPÍTULO VI

Conclusiones y Recomendaciones

Una vez desarrolladas cada una de las partes que componen el proyecto final de carrera, se presentan a continuación las conclusiones obtenidas luego de su realización, así como futuras recomendaciones para el proyecto.

VI.1.- Conclusiones

Se desarrolló una herramienta de validación e inventariado de routers que puede ejecutarse tanto de manera automática como manual. Disminuyendo la carga sobre recursos del proyecto, aumentando la eficiencia del mismo.

Se cumplieron los objetivos planteados, realizándolos además dentro del plazo de tiempo estimado inicialmente a pesar de las modificaciones de requerimientos e incidencias que fueron surgiendo.

Se desarrolló una página web de soporte para *PdS* desde la cual los técnicos introducen los datos de la instalación e inician su gestión.

Se crearon procesos de tratados de datos, mediante vistas y procedimientos SQL, entre la BBDD de la Web y la de desarrollos, así como cruces de tablas para la obtención completa de los datos de entrada necesarios a la herramienta del validador e inventariado de *routers*.

Se automatizó la carga de los inventarios de HP y NA a sus respectivas tablas de la BBDD de desarrollos, facilitando la comprobación de que los *routers* hayan sido registrados en dichos inventarios.



Se desarrolló un programa que realiza las validaciones para cada *router* aún validado, así como el registro de aquellos nuevos registros en el inventario de *CPEs*, automatizando dos procesos que se ejecutaban manualmente en el proyecto.

Se facilitó el proceso de post-instalación del proyecto a nivel general de manera que sea más fácil hacer la factura cada mes. Igualmente se disminuyó el tiempo medio del proceso de validación de los equipos instalados.

VII.2.- Recomendaciones

Entre las recomendaciones y mejoras propuestas para aumentar la utilidad de la herramienta se encuentran:

- Modificar la página web de soporte PdS para que se puedan introducir datos correspondientes a más de 2 routers por instalación, así como más de 2 accesos por router.
- Aumentar las funciones de la herramienta haciendo una parte que pueda automatizar la generación de la factura de manera que una vez validados los equipos se incluyen automáticamente en la siguiente factura cruzando con otros sistemas y bases de datos de las cuales se pueden obtener los precios a cobrar por cada ítem del proyecto, datos de SLAs y % de bonificaciones correspondiente según cantidad de ítems dentro del SLA en comparación con los que estuvieron fuera. Actualmente todo este proceso se sigue haciendo manualmente.
- Automatizar la subida de las plantillas de RFS a *confluence*, así como su envío por correo al jefe del proyecto. Actualmente este proceso se hace mediante macros de la plantilla excel que debe ejecutar manualmente el técnico después de generar la RFS desde la Web.



- Realizar una función que genere un resumen general de resultados de la ejecución del validador diariamente. Este resumen debería contener datos de cantidad total de routers pasados por el validador, cantidad total de validados, cantidad de fallos para cada evaluación, etc. Este documento puede ser de gran utilidad para tener un mayor control sobre los resultados de la herramienta y poder hacer históricos comparativos sobre su rendimiento.

BIBLIOGRAFÍA

- [1] Gómez Hernan, C (2010). Compilación bibliográfica-Business Process Outsourcing (BPO). Universidad de Caldas, Facultad de Ingenierías.
- [2] Ibermática (2015). Outsourcing de procesos de negocios (BPO).
- [3] Adecco (Mayo, 2015). VI Encuesta Adecco Outsourcing sobre Externalización.
- [4] Encarna González (2016) .Outsourcing tecnológico, una apuesta que gana adeptos en la empresa española.
- [5] Everis (2015). La externalización, cuestión de control y confianza.
- [6] González, I (2015). Provisión de Servicios: Procesos y Sistemas.
- [7] Microsoft Developer Network. Microsoft SQL Server.
- [8] Hewlett Packard Enterprise. Automatización de redes.
- [9] CISCO (2002). Manual Cisco CCNA - Las Interfaces de un Router.
- [10] Quesada, L (2010). Funcionamiento básico de CSS.
- [11] Hostalia (2015). Qué es el protocolo SSH y cómo configurarlo para mejorar la seguridad de acceso a los servidores Linux. Bilbao, Vizcaya.

Anexos

Anexo A

Código de la página web de soporte PDS

Sólo se muestran algunas funciones porque es un código demasiado extenso.

```
#Region "PROCEDIMIENTOS Y FUNCIONES"

'PROCEDIMIENTO GENÉRICO PARA PRESENTAR UN MENSAJE AL USUARIO
Protected Sub mensajeweb(mensaje As String)

    Dim sb As New System.Text.StringBuilder()

    sb.Append("<script type = 'text/javascript'>")

    sb.Append("window.onload=function(){")

    sb.Append("alert('")

    sb.Append(mensaje)

    sb.Append("');");

    sb.Append("</script>")

    ClientScript.RegisterClientScriptBlock(Me.GetType(), "alert", sb.ToString())

End Sub

'Procedimiento para guardar el adjunto
Private Sub SubirArchivo()

    Try
        'Si se ha seleccionado un archivo
        If AsyncFileUpload1.HasFile Then
            Dim wom As String = Me.txtWom.Text
            'se recoge el nombre y la ruta a guardar
            Dim filename As String =
System.IO.Path.GetFileName(AsyncFileUpload1.FileName)
            Dim rutadefinitiva As String
            Dim rutaTemp As String
            'Se guarda temporalmente en una subcarpeta
            rutaTemp = Server.MapPath("PDS/Archivos/Temp/") + filename
            rutadefinitiva = Server.MapPath("PDS/Archivos/") + filename
            ViewState("filename") = filename
            ViewState("rutatemp") = rutaTemp
            ViewState("rutadef") = rutadefinitiva
            'Se guarda
            AsyncFileUpload1.SaveAs(rutaTemp)
            'Se comprueba si el archivo está en la base de datos
```



```
'No puede haber dos archivos iguales en la misma carpeta
Dim sql As String = "SELECT [NOMBRE_ARCHIVO] FROM
[orange_ggcc_bo].[pds].[PDS_ACTIVIDAD_ADJUNTOS] WHERE [NOMBRE_ARCHIVO] = '" & filename & "'"
If consultasSQL(sql) <> "" Then
'Si ya existe se presenta una ventana de confirmación
ModalPopupExtender1.Show()
Else
If System.IO.File.Exists(rutadefinitiva) = True Then
System.IO.File.Delete(rutadefinitiva)
End If
'Si no existe se mueve a la carpeta definitiva
System.IO.File.Move(rutaTemp, rutadefinitiva)
'Se copia a Google Drive
If Left(filename, 7) = wom & "_" And (Right(filename, 4) = ".xls" Or
Right(filename, 3) = ".txt") Then
CopiarArchivoaGDrive()
End If
'Se insertan los datos en lla base de datos
InsertarAdjunto(txtWom.Text, filename, txtDescripcionArchivo.Text,
rutadefinitiva)
'Se actualiza el grid de archivos y se limpia el cuadro de texto de
descripción
rellenargridarchivos()
txtDescripcionArchivo.Text = ""
Dim mensaje As String = "Archivo subido"
ScriptManager.RegisterStartupScript(Me.Page, Me.GetType(), "Popup2",
>alert('" & mensaje & "');", True)
logTxt(Now() & " - Web CdMProvisión - Subido archivo soporte PDS")
End If
'Si no se ha seleccionado ningún archivo
Else
Dim mensaje As String = "Por favor, seleccione un archivo"
ScriptManager.RegisterStartupScript(Me.Page, Me.GetType(), "Popup2",
>alert('" & mensaje & "');", True)
End If

Catch ex As Exception
Dim mensaje As String = "Error: " & ex.Message & " Por favor, actualice la
página y vuelva a intentarlo"
ScriptManager.RegisterStartupScript(Me.Page, Me.GetType(), "Popup2", "alert('" &
mensaje & "');", True)
logTxt(Now() & " - Web CdMProvisión - ERROR al subir archivo soporte PDS")
End Try

End Sub

'Procedimiento para subir el archivo de Excel de CPE
Private Sub SubirArchivoCPE(nwom As Integer)

Try
'Si se ha seleccionado un archivo
If AsyncFileUpload2.HasFile Then
'se recoge el nombre y la ruta a guardar
Dim filename As String =
System.IO.Path.GetFileName(AsyncFileUpload2.FileName)
Dim rutaTemp As String
'Se guarda temporalmente en la subcarpeta Temp
rutaTemp = Server.MapPath("ExcelSubidos/") + filename
'Se guarda
AsyncFileUpload2.SaveAs(rutaTemp)
'LEE LOS DATOS Y LOS ACTUALIZA
If leerExcelCPE(Server.MapPath("ExcelSubidos/") + filename, nwom) Then
'SUBIRLO A LA CARPETA DE ARCHIVOS
If System.IO.File.Exists(Server.MapPath("PDS/Archivos/") + filename) =
True Then
```



```
System.IO.File.Delete(Server.MapPath("PDS/Archivos/") + filename)
End If
'COPIAR A LA CARPETA DE ARCHIVOS
filename) System.IO.File.Copy(rutaTemp, Server.MapPath("PDS/Archivos/") +
'COPIAR A GOOGLE DRIVE
If Left(filename, 7) = nwom & "_" And Right(filename, 4) = "xlsm" Then
ViewState("filename") = filename
ViewState("rutadef") = Server.MapPath("PDS/Archivos/") + filename
CopiarArchivoaGDrive()
End If
'INSERTAR REGISTRO EN LA BASE DE DATOS
Server.MapPath("PDS/Archivos/") + filename) InsertarAdjunto(txtWom.Text, filename, txtDescripcionArchivo.Text,
descripción) 'Se actualiza el grid de archivos y se limpia el cuadro de texto de
'PRESENTA EL MENSAJE AL USUARIO
Dim mensaje As String = "ARCHIVO PROCESADO"
ScriptManager.RegisterStartupScript(Me.Page, Me.GetType(), "Popup2",
>alert('" & mensaje & "');" , True)
'ELIMINAR EL ARCHIVO DE EXCEL EN EL SERVIDOR
System.IO.File.Delete(Server.MapPath("ExcelSubidos/") + filename)
'VUELVE A ACTUALIZAR EL GRID DE CPES Y DE ARCHIVOS
rellenargridCPES(nwom)
rellenargridarchivos()
Else
Dim mensaje As String = "ERROR AL PROCESAR EL ARCHIVO, NO TIENE EL
FORMATO CORRECTO O NO CORRESPONDE A ESTE Nº WOM"
ScriptManager.RegisterStartupScript(Me.Page, Me.GetType(), "Popup2",
>alert('" & mensaje & "');" , True)
End If
Else
Dim mensaje As String = "Por favor, seleccione un archivo"
ScriptManager.RegisterStartupScript(Me.Page, Me.GetType(), "Popup2",
>alert('" & mensaje & "');" , True)
End If

Catch ex As Exception
Dim mensaje As String = "Error: " & ex.Message & " Por favor, actualice la
página y vuelva a intentarlo"
ScriptManager.RegisterStartupScript(Me.Page, Me.GetType(), "Popup2", "alert('" &
mensaje & "');" , True)
logTxt(Now() & " - Web CdMProvisión - ERROR al subir plantilla Excel CPE de
soporte PDS")

End Try

End Sub

'Procedimiento que inserta la información de comentarios en la base de datos
Private Sub InsertarComentario(IdWom As Integer, coment As String, tipo As String)
Try

Dim sql As String = "INSERT INTO
[orange_ggcc_bo].[ggcc].GGCC_ACTIVIDAD_COMENTARIOS (WOM_PROYECTO, COMENTARIO,
FECHA_COMENTARIO, ACTOR, TIPO_COMENTARIO) VALUES(" & IdWom & " ,'" & coment & "' , '" &
dateToSQLString(DateTime.Now) & "' , 'PDS'" & " , '" & tipo & "'" "
ejecutarcomando(sql)

Catch ex As Exception
Dim mensaje As String = "Error al añadir comentario: " & ex.Message
ScriptManager.RegisterStartupScript(Me.Page, Me.GetType(), "Popup2", "alert('" &
mensaje & "');" , True)

End Try
```



```
End Sub

'Procedimiento que inserta la información de los archivos adjuntos en la base de datos
Private Sub InsertarAdjunto(IdWom As Integer, nombre As String, descripcion As String,
ruta As String)
    Try
        Dim sql As String = "INSERT INTO [orange_ggcc_bo].[pds].[PDS_ACTIVIDAD_ADJUNTOS]
(WOM_PROYECTO, NOMBRE_ARCHIVO, DESCRIPCION_ARCHIVO,RUTA_ARCHIVO, FECHA_ADJUNTO) VALUES(" &
IdWom & " ,'" & nombre & "' ,'" & descripcion & "' ,'" & ruta & "' ,'" &
dateToSQLString(DateTime.Now) & "' ) "
        ejecutarcomando(sql)

    Catch ex As Exception
        Dim mensaje As String = "Error al insertar los datos del archivo: " & ex.Message
        ScriptManager.RegisterStartupScript(Me.Page, Me.GetType(), "Popup2", "alert('" &
mensaje & "');", True)

    End Try

End Sub

'Procedimiento que elimina la información de los adjuntos sustituidos en la base de
datos
Private Sub EliminarAdjunto(archivo As String)
    Dim ssql As String
    ssql = "DELETE FROM [orange_ggcc_bo].pds.PDS_ACTIVIDAD_ADJUNTOS WHERE NOMBRE_ARCHIVO
= '" & archivo & "'"
    ejecutarcomando(ssql)

    Try

    Catch ex As Exception
        Dim mensaje As String = "Error al eliminar el registro del archivo: " &
ex.Message
        ScriptManager.RegisterStartupScript(Me.Page, Me.GetType(), "Popup2", "alert('" &
mensaje & "');", True)

    End Try
End Sub

'procedimiento para eliminar el archivo desde el grid de archivos
Protected Sub EliminarArchivo(ByVal sender As Object, ByVal e As EventArgs)
    Dim archivo As String
    Dim ssql As String
    Dim row As GridViewRow
    Dim grid As GridView
    Dim wom As String
    Dim ruta As String

    Try

        row = CType(CType(sender, ImageButton).Parent.Parent, GridViewRow)
        grid = CType(UpdatePanel3.FindControl(row.Parent.Parent.ID), GridView)
        'busca el nombre del archivo de la fila correspondiente
        archivo = grid.DataKeys(row.RowIndex).Value.ToString()
        wom = Me.txtWom.Text
        'borra el registro de la base de datos
        ssql = "DELETE FROM [orange_ggcc_bo].pds.PDS_ACTIVIDAD_ADJUNTOS WHERE
NOMBRE_ARCHIVO = '" & archivo & "' AND WOM_PROYECTO=" & wom
        ejecutarcomando(ssql)
        ruta = Server.MapPath("PDS/Archivos/") + archivo
        'elimina el archivo de la carpeta del servidor
        If System.IO.File.Exists(ruta) = True Then
            System.IO.File.Delete(ruta)
        End If
    End Try
End Sub
```



```
End If
rellenargridarchivos()
Catch ex As Exception
Dim mensaje As String = "Error al eliminar el archivo: " & ex.Message
ScriptManager.RegisterStartupScript(Me.Page, Me.GetType(), "Popup2", "alert('" &
mensaje & "');" , True)
End Try

End Sub

'Eliminar comentarios
Protected Sub EliminarComentario(ByVal sender As Object, ByVal e As EventArgs)
Dim n As String
Dim ssql As String
Dim row As GridViewRow
Dim grid As GridView
'Dim wom As String

Try

row = CType(CType(sender, ImageButton).Parent.Parent, GridViewRow)
grid = CType(UpdatePanel2.FindControl(row.Parent.Parent.ID), GridView)
n = grid.DataKeys(row.RowIndex).Value.ToString()
'wom = Me.txtWom.Text
ssql = "DELETE FROM [orange_ggcc_bo].[ggcc].[GGCC_ACTIVIDAD_COMENTARIOS] WHERE
[ID]= '" & n & "'"
ejecutarcomando(ssql)
rellenargridcomentarios()
Catch ex As Exception
Dim mensaje As String = "Error al eliminar el comentario: " & ex.Message
ScriptManager.RegisterStartupScript(Me.Page, Me.GetType(), "Popup2", "alert('" &
mensaje & "');" , True)

End Try
End Sub

'Procedimiento para eliminar los datos técnicos de los CPES
Protected Sub EliminarCPE(ByVal sender As Object, ByVal e As EventArgs)
Dim hostname As String
Dim ssql As String
Dim row As GridViewRow
Dim grid As GridView
Dim wom As String

Try

row = CType(CType(sender, ImageButton).Parent.Parent, GridViewRow)
grid = CType(UpdatePanel6.FindControl(row.Parent.Parent.ID), GridView)
hostname = grid.DataKeys(row.RowIndex).Value.ToString()
wom = Me.txtWom.Text
ssql = "DELETE FROM [orange_ggcc_bo].[pds].[PDS_ACTIVIDAD_CPES] WHERE
[WOM_PROYECTO]= '" & wom & "' AND [HOSTNAME]= '" & hostname & "'"
ejecutarcomando(ssql)
rellenargridCPES(wom)
Catch ex As Exception
Dim mensaje As String = "Error al eliminar los datos del CPE: " & ex.Message
ScriptManager.RegisterStartupScript(Me.Page, Me.GetType(), "Popup2", "alert('" &
mensaje & "');" , True)

End Try
End Sub

'Procedimiento que inserta la información de la instalación
```



DISEÑO E IMPLEMENTACIÓN DE UNA HERRAMIENTA DE VALIDACIÓN E INVENTARIADO DE ROUTERS

```
Private Sub InsertarInstalación(wom As Integer, ffingestion As String, hp As String,
tacas As String, idocean As String, na As String, jefe As String)
    Try
        Dim sql As String = "INSERT INTO
[orange_ggcc_bo].[pds].[PDS_ACTIVIDAD_INSTALACIONES] (WOM_PROYECTO, FECHA_FIN_GESTION, HP,
TACAS, ID_OCEANE, NA, JEFE_PROYECTO) " & _
        " VALUES(" & wom & ", " & ffingestion & ", " & hp & ", " &
tacas & ", " & idocean & ", " & na & ", " & jefe & " )"
        ejecutarcomando(sql)

    Catch ex As Exception
        Dim mensaje As String = "Error al añadir comentario: " & ex.Message
        ScriptManager.RegisterStartupScript(Me.Page, Me.GetType(), "Popup2", "alert('" &
mensaje & "');", True)

    End Try

End Sub
```



Anexo B

Diseño tabla de Validaciones IS2_IMPORTED_DATA

Nombre columna	Tipo de dato	Nivel de dato	Origen
ID	int	Registro tabla	automático
[Código RFS]	nvarchar(MAX)	instalación	BBDD Web PdS
[Numero de Visita]	int	instalación	BBDD Web PdS
[Fecha Inicio]	datetime	instalación	BBDD Web PdS
[Fecha Fin]	datetime	instalación	BBDD Web PdS
Estado	nvarchar(50)	instalación	BBDD Web PdS
[Fecha Última	datetime	instalación	BBDD Web PdS
WOM	float	instalación	BBDD Web PdS
ITEM	nvarchar(50)	instalación	BBDD Web PdS
JP	nvarchar(50)	instalación	BBDD Web PdS
CLIENTE	nvarchar(MAX)	instalación	BBDD Web PdS
[TIPO ACCIÓN]	nvarchar(50)	instalación	BBDD Web PdS
[WOM ALTA]	float	instalación	BBDD Web PdS
Red	nvarchar(50)	instalación	BBDD Web PdS
[Número de CPEs]	int	instalación	BBDD Web PdS
ID_CPE	int	equipo	BBDD Web PdS
[Código CPE]	nvarchar(MAX)	equipo	BBDD Web PdS
[Número de Accesos]	int	Equipo	BBDD Web PdS
ID_Acceso	int	acceso	BBDD Web PdS
[Tipo Servicio]	nvarchar(MAX)	acceso	BBDD Web PdS
Tecnología	nvarchar(MAX)	acceso	BBDD Web PdS
Router	nvarchar(MAX)	equipo	BBDD Web PdS
PE	nvarchar(MAX)	equipo	BBDD Web PdS
[Código Línea]	nvarchar(MAX)	acceso	BBDD Web PdS
Router_cliente	nvarchar(MAX)	equipo	BBDD Web PdS
[Orden Nemonico]	nvarchar(MAX)	equipo	BBDD Web PdS
Tipoderouter	nvarchar(MAX)	equipo	BBDD Web PdS
Concentrador	nvarchar(MAX)	equipo	BBDD Web PdS
[Interfaz concentrador]	nvarchar(MAX)	equipo	BBDD Web PdS
IP_loopback	nvarchar(MAX)	equipo	BBDD Web PdS
[Subred conexión]	nvarchar(MAX)	equipo	BBDD Web PdS
DireccionWAN_RC	nvarchar(MAX)	equipo	BBDD Web PdS
DireccionWAN_CPE	nvarchar(MAX)	equipo	BBDD Web PdS
RD	nvarchar(MAX)	equipo	BBDD Web PdS
VRF	nvarchar(MAX)	equipo	BBDD Web PdS
IOS	nvarchar(MAX)	equipo	BBDD Web PdS



DISEÑO E IMPLEMENTACIÓN DE UNA HERRAMIENTA DE VALIDACIÓN E INVENTARIADO DE ROUTERS

Nombre columna	Tipo de dato	Nivel de dato	Origen
QoS	nvarchar(MAX)	equipo	BBDD Web PdS
SerialRouter	nvarchar(MAX)	equipo	BBDD Web PdS
[Direccionamiento	nvarchar(MAX)	equipo	BBDD Web PdS
[Direccionamiento	nvarchar(MAX)	equipo	BBDD Web PdS
AccUsuari	nvarchar(MAX)	equipo	BBDD Web PdS
AccDomini	nvarchar(MAX)	equipo	BBDD Web PdS
[Tecnologia de Acceso]	nvarchar(50)	acceso	BBDD Web PdS
[Ancho de Banda]	nvarchar(50)	acceso	BBDD Web PdS
Modalidad	nvarchar(50)	acceso	BBDD Web PdS
[Numero	nvarchar(50)	acceso	BBDD Web PdS
[Código de Acceso]	nvarchar(50)	acceso	BBDD Web PdS
HP	bit	instalación	BBDD Web PdS
NA	bit	instalación	BBDD Web PdS
TACACS	bit	instalación	BBDD Web PdS
ID Oceane	nvarchar(MAX)	instalación	BBDD Web PdS
Responsable_Instalacion	nvarchar(MAX)	instalación	BBDD Web PdS
[Fecha Fin Gestion]	datetime	instalación	BBDD Web PdS
[Certificación Técnico]	bit	instalación	BBDD Web PdS
[Router Activo]	bit	instalación	BBDD Web PdS
[Pruebas LAN]	bit	instalación	BBDD Web PdS
Observaciones	nvarchar(MAX)	instalación	BBDD Web PdS
[Validador Cliente]	nvarchar(MAX)	instalación	BBDD Web PdS
[Asistencia técnica]	bit	instalación	BBDD Web PdS
[Asistente técnico]	nvarchar(MAX)	instalación	BBDD Web PdS
[LAN conectada]	bit	instalación	BBDD Web PdS
[Motivo LAN	nvarchar(MAX)	instalación	BBDD Web PdS
Check_OBS	bit	instalación	BBDD Web PdS
OBS_Interconexion	nvarchar(MAX)	instalación	BBDD Web PdS
OBS_VLAN	nvarchar(MAX)	instalación	BBDD Web PdS
[Motivo KO]	nvarchar(MAX)	instalación	BBDD Web PdS
[Número de Incidencia]	nvarchar(50)	instalación	BBDD Web PdS
Comentario	nvarchar(MAX)	instalación	BBDD Web PdS
[Fecha última Validación	datetime	registro	Validador
[Iteración de validación]	int	registro	Validador
[Acceso IP]	bit	registro	Validador
[Configuración DNS]	bit	registro	Validador
[Alta TACACS]	bit	registro	Validador
[Alta HP Inventory]	bit	registro	Validador
[Alta NA Inventory]	bit	registro	Validador
[OK Tareas WOM]	bit	registro	Validador
[Resultado Final]	bit	registro	Validador
[Tratar RFS]	bit	registro	Validador



Nombre columna	Tipo de dato	Nivel de dato	Origen
[Correo enviado a JP]	bit	registro	Validador
[Campos para	nvarchar(MAX)	registro	Validador
[RFS en Confluence?]	nchar(10)	registro	Validador
[Proyecto Facturado?]	nchar(10)	registro	Validador
[Cantidad KOs]	int	registro	Validador

Anexo C

Diseño tabla de Inventario_CPEs_DEF

Nombre del campo	Tabla de origen
ID	PK Autoincrementable
IS2_ID	Validaciones
[Fecha Registro]	Fecha de inventariado
[Fecha Ultima Modificacion]	Validaciones
Red	Validaciones
[Tipo Acción]	Validaciones
[Wom Alta]	Validaciones
[Código RFS]	Validaciones
[Datosadministrativosdecliente.Nombre]	Validaciones
DirecciónSede	WOM
Router_cliente	Validaciones
OrdenNemonico	Validaciones
Tipoderouter	Validaciones
Nombre_RouterConcentrador	Validaciones
Interfazconcentrador	Validaciones
IP_loopback	Validaciones
Subredeconexión	Validaciones
DireccionWAN_RC	Validaciones
DireccionWAN_CPE	Validaciones
RD	Validaciones
VRF	Validaciones
[Código/sLinea/s]	Validaciones
Tecnología	Validaciones
AnchodeBanda	Validaciones
NúmeroadministrativooRTB	Validaciones
IOS	Validaciones
QoS	Validaciones



Nombre del campo	Tabla de origen
SerialRouter	Validaciones
[DireccionamientoPrivado(sede)]	Validaciones
DireccionamientoPublicas	Validaciones
Nombrecontactocliente	Gescir
Teléfonocontactocliente	Gescir
JPC	Validaciones
Códigodelcliente	WOM
WOM	Validaciones
Servicio	WOM
AccUsuari	Validaciones
AccDomini	Validaciones
Comentarios	Validaciones

Anexo D

Vista SQL CARGA_WEB_PDS_SQL

```

SELECT *
FROM (SELECT DISTINCT
RFS.ID_RFS AS [Código RFS], VI.NUM_VISITA AS [Numero de Visita],
VI.FECHA_INICIO AS [Fecha Inicio], VI.FECHA_FIN AS [Fecha Fin], VI.ESTADO AS
[Estado],
GETDATE() AS [Fecha Última Modificación], REP.WOM_PROYECTO AS [WOM],
REP.CODIGO_ITEM AS [ITEM], REP.JEFE_PROYECTO AS [JP],
REP.CLIENTE AS [CLIENTE], REP.TIPO_MOVIMIENTO AS [TIPO ACCIÓN], (CASE WHEN
REP.WOM_ALTA IS NOT NULL AND
REP.WOM_ALTA <> '' THEN REP.WOM_ALTA ELSE ALTA.WOM_ALTA END) AS [WOM ALTA],
ACC.RED AS [Red],
(SELECT TOP 1 R.ID_CPEs
FROM [orange_ggcc_bo].[pds].[PDS_ACTIVIDAD_RFS] AS R
WHERE R.WOM_PROYECTO = REP.WOM_PROYECTO
ORDER BY R.ID_CPEs DESC) AS [Número de CPEs], RFS.ID_CPEs AS [ID_CPE],
RFS.COD_CPE AS [Código CPE], RFS.N_ACCESOS AS [Número de Accesos],
ACC.ID_ACCESO AS [ID_Acceso], RFS.TIPO_SERVICIO AS [Tipo Servicio],
RFS.TECNOLOGIA AS [Tecnología], RFS.ROUTER AS [Router], RFS.PE AS [PE],
CPE.CODIGO_LINEA AS [Código Línea], CPE.HOSTNAME AS [Router_cliente],
CPE.ORDEN_NEMONICO AS [Orden Nemonico],
CPE.TIPO_ROUTER AS [Tipoderouter], CPE.ROUTER_CONCENTRADOR AS [Concentrador],
CPE.INTERFAZ_CONCENTRADOR AS [Interfaz concentrador],
CPE.IP_LOOPBACK AS [IP_loopback], CPE.SUBRED_CONEXION AS [Subred conexión],
CPE.WAN_RC AS [DireccionWAN_RC],
CPE.WAN_CPE AS [DireccionWAN_CPE], CPE.RD AS [RD], CPE.VRF AS [VRF], CPE.IOS AS
[IOS], CPE.QoS AS [QoS], CPE.SERIAL_ROUTER AS [SerialRouter],
CPE.DIRECC_PRIVADO AS [Direccionamiento Privado(sede)], CPE.DIRECC_PUBLICAS AS
[Direccionamiento Publicas], CPE.AAC_USUARIO AS [AccUsuari],
CPE.AAC_DOMINIO AS [AccDomini], ACC.TECNOLOGIA AS [Tecnología de Acceso],
ACC.ANCHO_BANDA AS [Ancho de Banda], ACC.MODALIDAD AS [Modalidad],

```



```
ACC.NUMERO_ADMINISTRATIVO AS [Numero Administrativo], ACC.CODIGO_ACCESO AS
[Código de Acceso], INST.HP AS [HP], INST.NA AS [NA],
INST.TACAS AS [TACACS], INST.ID_OCEANE AS [ID_Oceane], INST.RESP_INSTALACION AS
[Responsable_Instalacion],
INST.FECHA_FIN_GESTION AS [Fecha Fin Gestion], INST.CERTIFICACION_TECNICO AS
[Certificación Técnico], INST.ROUTER_ACTIVADO AS [Router Activo],
INST.PRUEBAS_LAN AS [Pruebas LAN], INST.OBSERVACIONES_RFS AS [Observaciones],
INST.VALIDADOR_CLIENTE AS [Validador Cliente],
INST.ASISTENCIA_TECNICA AS [Asistencia técnica], INST.ASISTENTE_TECNICO AS
[Asistente técnico], INST.LAN_CONECTADA AS [LAN conectada],
INST.MOTIVO_LAN_DESCONECTADA AS [Motivo LAN desconectada], INST.CHECK_OBS AS
[Check_OBS], INST.OBS_INTERCONEXION AS [OBS_Interconexion],
INST.OBS_VLAN AS [OBS_VLAN], VI.MOTIVO_KO AS [Motivo KO], VI.NUM_INCIDENCIA AS
[Numero de Incidencia], VI.COMENTARIO AS [Comentario],
'' AS [Fecha última Validación gestión equipo],
(SELECT count(I.WOM)
FROM [orange_cdm_industrializado].[dbo].[IS2_IMPORTED_DATA_DEF] AS I
WHERE I.WOM = rep.WOM_PROYECTO AND I.[TIPO ACCIÓN] = REP.[TIPO MOVIMIENTO]
AND I.ID_CPE = ID_CPE AND I.ID_Acceso = ID_Acceso AND
I.[Tecnología de Acceso] = [Tecnología de Acceso]) AS NUM_REPETIDOS
FROM [orange_cdm_ggcc].[carga].[PROYECTOS_REPOSITORIO_UNICO] AS REP LEFT
OUTER JOIN
[orange_cdm_ggcc].[dbo].[WOM_ALTA] AS ALTA ON REP.WOM_PROYECTO = ALTA.WOM LEFT
OUTER JOIN
[orange_ggcc_bo].[pds].PDS_ACTIVIDAD_INSTALACIONES AS INST ON REP.WOM_PROYECTO
= INST.WOM_PROYECTO LEFT OUTER JOIN
(SELECT WOM_PROYECTO, NUM_VISITA, FECHA_INICIO, FECHA_FIN, ESTADO, MOTIVO_KO,
NUM_INCIDENCIA, COMENTARIO
FROM (SELECT rank() OVER (partition BY v.wom_proyecto
ORDER BY v.NUM_VISITA DESC) AS VI_RANK, v.wom_proyecto, v.NUM_VISITA,
v.FECHA_INICIO, v.FECHA_FIN, v.ESTADO, v.MOTIVO_KO,
v.NUM_INCIDENCIA, v.COMENTARIO
FROM [orange_ggcc_bo].[pds].PDS_ACTIVIDAD_INSTALACIONES_VISITAS AS v) AS
VIS
WHERE VIS.vi_rank = 1) AS VI ON REP.WOM_PROYECTO = VI.WOM_PROYECTO LEFT OUTER
JOIN
[orange_ggcc_bo].[pds].PDS_ACTIVIDAD_CPES AS CPE ON REP.WOM_PROYECTO =
CPE.WOM_PROYECTO LEFT OUTER JOIN
[orange_ggcc_bo].[pds].PDS_ACTIVIDAD_ACCESOS AS ACC ON CPE.WOM_PROYECTO =
ACC.WOM_PROYECTO AND
CPE.CODIGO_LINEA = ACC.CODIGO_ACCESO LEFT OUTER JOIN
[orange_ggcc_bo].[pds].[PDS_ACTIVIDAD_RFS] AS RFS ON CPE.WOM_PROYECTO =
RFS.WOM_PROYECTO AND CPE.HOSTNAME = RFS.COD_CPE) AS A
WHERE (SELECT count(I.WOM)
FROM [orange_cdm_industrializado].[dbo].[IS2_IMPORTED_DATA_DEF] AS I
WHERE (I.WOM = A.WOM OR
I.WOM IS NULL) AND (I.[TIPO ACCIÓN] = A.[TIPO ACCIÓN] OR
I.[TIPO ACCIÓN] IS NULL) AND (I.ID_CPE = A.ID_CPE OR
I.ID_CPE IS NULL) AND (I.ID_Acceso = A.ID_Acceso OR
I.ID_Acceso IS NULL) AND (I.[Tecnología de Acceso] = A.[Tecnología de Acceso]
OR
I.[Tecnología de Acceso] IS NULL)) = 0 AND A.ESTADO = 'OK'
```



Anexo E

Procedimiento SQL INSERT_INTO_IS2

```
USE [orange_cdm_industrializado]
GO
```

```
/****** Object: StoredProcedure [dbo].[INSERT_INTO_IS2]    Script Date:
24/10/2016 11:09:03 *****/
```

```
SET ANSI_NULLS ON
GO
```

```
SET QUOTED_IDENTIFIER ON
GO
```

```
CREATE PROCEDURE [dbo].[INSERT_INTO_IS2]
AS
BEGIN
```

```
INSERT INTO IS2_IMPORTED_DATA_DEF
    ([Código RFS]
    ,[Numero de Visita]
    ,[Fecha Inicio]
    ,[Fecha Fin]
    ,[Estado]
    ,[Fecha Última Modificación]
    ,[WOM]
    ,[ITEM]
    ,[JP]
    ,[CLIENTE]
    ,[TIPO ACCIÓN]
    ,[WOM ALTA]
    ,[Red]
    ,[Número de CPEs]
    ,[ID_CPE]
    ,[Código CPE]
    ,[Número de Accesos]
    ,[ID_Acceso]
    ,[Tipo Servicio]
    ,[Tecnología]
    ,[Router]
    ,[PE]
    ,[Código Línea]
    ,[Router_cliente]
    ,[Orden Nemonico]
    ,[Tipoderouter]
    ,[Concentrador]
    ,[Interfaz concentrador]
    ,[IP_loopback]
    ,[Subred conexión]
    ,[DireccionWAN_RC]
```



```
,[DireccionWAN_CPE]
,[RD]
,[VRF]
,[IOS]
,[QoS]
,[SerialRouter]
,[Direccionamiento Privado(sede)]
,[Direccionamiento Publicas]
,[AccUsuari]
,[AccDomini]
,[Tecnologia de Acceso]
,[Ancho de Banda]
,[Modalidad]
,[Numero Administrativo]
,[Código de Acceso]
,[HP]
,[NA]
,[TACACS]
,[ID_Oceane]
,[Responsable_Instalacion]
,[Fecha Fin Gestion]
,[Certificación Técnico]
,[Router Activo]
,[Pruebas LAN]
,[Observaciones]
,[Validador Cliente]
,[Asistencia técnica]
,[Asistente técnico]
,[LAN conectada]
,[Motivo LAN desconectada]
,[Check_OBS]
,[OBS_Interconexion]
,[OBS_VLAN]
,[Motivo KO]
,[Numero de Incidencia]
,[Comentario]
,[Fecha última Validación gestión equipo])
  (SELECT [Código RFS]
,[Numero de Visita]
,[Fecha Inicio]
,[Fecha Fin]
,[Estado]
,[Fecha Última Modificación]
,[WOM]
,[ITEM]
,[JP]
,[CLIENTE]
,[TIPO ACCIÓN]
,[WOM ALTA]
,[Red]
,[Número de CPEs]
,[ID_CPE]
,[Código CPE]
,[Número de Accesos]
,[ID_Acceso]
```



```
,[Tipo Servicio]
,[Tecnología]
,[Router]
,[PE]
,[Código Linea]
,[Router_cliente]
,[Orden Nemonico]
,[Tipoderouter]
,[Concentrador]
,[Interfaz concentrador]
,[IP_loopback]
,[Subred conexión]
,[DireccionWAN_RC]
,[DireccionWAN_CPE]
,[RD]
,[VRF]
,[IOS]
,[QoS]
,[SerialRouter]
,[Direccionamiento Privado(sede)]
,[Direccionamiento Publicas]
,[AccUsuari]
,[AccDomini]
,[Tecnologia de Acceso]
,[Ancho de Banda]
,[Modalidad]
,[Numero Administrativo]
,[Código de Acceso]
,[HP]
,[NA]
,[TACACS]
,[ID_Oceane]
,[Responsable_Instalacion]
,[Fecha Fin Gestion]
,[Certificación Técnico]
,[Router Activo]
,[Pruebas LAN]
,[Observaciones]
,[Validador Cliente]
,[Asistencia técnica]
,[Asistente técnico]
,[LAN conectada]
,[Motivo LAN desconectada]
,[Check_OBS]
,[OBS_Interconexion]
,[OBS_VLAN]
,[Motivo KO]
,[Numero de Incidencia]
,[Comentario]
,[Fecha última Validación gestión equipo]
  FROM [dbo].[CARGA_PDS_WEB_SQL])
```

END

GO



Anexo F

Código CargaNA.vb

IELanSettings.vb

```
Imports Microsoft.VisualBasic
Imports Microsoft.Win32

Public Class IELanSettings
    Dim Registro As RegistryKey =
Registry.CurrentUser.CreateSubKey("Software\Microsoft\Windows\CurrentVersion\Internet
Settings", RegistryKeyPermissionCheck.ReadSubTree)
    Dim EditReg As RegistryKey =
Registry.CurrentUser.CreateSubKey("Software\Microsoft\Windows\CurrentVersion\Internet
Settings", RegistryKeyPermissionCheck.ReadWriteSubTree)

    Public Sub New()

    End Sub

    Public Function IsAutoScriptEnable() As Boolean
    Try
        If Me.Registro.GetValue("AutoConfigURL") <> Nothing Then
            Return True
        Else
            Return False
        End If
    Catch ex As Exception
        MsgBox("Error: " + ex.Message)
        Return False
    End Try
    End Function

    Public Function getAutomaticConfigurationScript() As String
    Try
        If Me.Registro.GetValue("AutoConfigURL") <> Nothing Then
            Return Me.Registro.GetValue("AutoConfigURL").ToString
        Else
            Return ""
        End If
    Catch ex As Exception
        MsgBox("Error: " + ex.Message)
        Return ("ERROR")
    End Try
    End Function

    Public Function UncheckAutoScript() As Boolean
    Try
        Me.EditReg.SetValue("AutoConfigURL", "").ToString, RegistryValueKind.String)
        Return True
    Catch ex As Exception
        Return False
    End Try
    End Function

    Public Function setAutomaticConfigurationScript(script As String) As Boolean
    Try
```



```
        Me.EditReg.SetValue("AutoConfigURL", script, RegistryValueKind.String)
        Return True
    Catch ex As Exception
        Return False
    End Try
End Function

Public Function IsProxyEnable() As Boolean
    Try
        If Me.Registro.GetValue("ProxyEnable") <> Nothing Then
            If Me.Registro.GetValue("ProxyEnable").ToString = "0" Then
                Return False
            Else
                Return True
            End If
        End If

    Catch ex As Exception
        MsgBox("Error: " + ex.Message)
        Return False
    End Try
    Return False
End Function

Public Function getProxy() As String
    Try
        If Me.Registro.GetValue("ProxyServer") <> Nothing Then
            Return Me.Registro.GetValue("ProxyServer").ToString
        Else
            Return ""
        End If
    Catch ex As Exception
        MsgBox("Error: " + ex.Message)
        Return ("ERROR")
    End Try
End Function

Public Function setProxy(proxyState As Boolean, proxyserver As String) As Boolean
    Try
        Me.EditReg.SetValue("ProxyEnable", proxyState, RegistryValueKind.DWord)
        Me.EditReg.SetValue("ProxyServer", proxyserver, RegistryValueKind.String)
        Return True
    Catch ex As Exception
        Return False
    End Try
End Function

End Class
```

ControllerNAInfo.vb

```
Imports HelioGetDataBySelenium
Imports SeleniumBase
Imports Utils.Utils

Namespace Controllers
    'Controlador que se encarga de trabajar con la pagina NA
    Public Class ControllerNAInfo
        Private mObjNA As NAAccess
        Private mObjDriver As DriverBase
        Private NAUsername As String
        Private NAPassword As String
    End Class
End Namespace
```



```
Public Sub New(ByVal username As String, ByVal password As String)
    Try
        NAUsername = username
        NAPassword = password
        StartNA()
    Catch ex As Exception

    End Try
End Sub

    Try
        'Se crea un nuevo webdriver de internet explorer
        mObjDriver = New DriverBase(DriverBase.WebDriverType.IE)

        'Se crea un nuevo objeto de NA
        mObjNA = New NAAccess(mObjDriver)

        mObjNA.ConnectToNA()
        mObjNA.ByPassCertificateMessage()
        mObjNA.TryToLoginNA(NAUsername, NAPassword)
        mObjNA.ConnectToNA()

    Catch ex As Exception
        Throw New Exception(ex.Message)
    End Try
End Sub

Public Function getSourceCodeRaw() As String
    Try
        Dim datosRaw As String
        datosRaw = mObjNA.getRoutersInfo()
        Return datosRaw
    Catch ex As Exception
        Return ""
    End Try
End Function

Private Sub RestartNA()
    mObjDriver.Dispose()
    mObjNA.Dispose()
    StartNA()
End Sub

Public Sub Dispose()
    mObjDriver.Dispose()
    mObjNA.Dispose()
    mObjDriver = Nothing
    mObjNA = Nothing
End Sub

End Class
End Namespace
```

NAModel.vb

```
Imports DatabaseAbstractionLayer

Namespace Models
    Public Class NAModel
        Private Property hostname As String
```



```
Private Property loopback As String

Public Sub New(ByVal dr As DataRow)
    Try
        Me.hostname = CStr(dr("hostname"))
        Me.loopback = CStr(dr("loopback"))
    Catch ex As Exception
        Throw New Exception(ex.Message & "ROW_VALUES: " &
Utils.Utils.ConvertDataRowValuesAsString(dr))
    End Try
End Sub

Public Sub InsertToDB(ByVal objConnection As SQLServerDataConnection, ByVal tbl_name
As String)
    Dim sqlQuery As String = "INSERT " & tbl_name & "(HOSTNAME, LOOPBACK) VALUES
(@hostname, @loopback)"

    Dim cmd As New SqlCommand(sqlQuery, objConnection._connection)
    cmd.Parameters.Add(New SqlParameter("@hostname", Me.hostname))
    cmd.Parameters.Add(New SqlParameter("@loopback", Me.loopback))

    Try
        objConnection.ExecuteNonQuery(cmd)
    Catch ex As Exception
        Throw New Exception(ex.Message & "ROW_VALUES: " &
Utils.Utils.ConvertParametersValuesAsString(cmd))
    End Try
End Sub

End Class
End Namespace
```

NAAcces.vb

```
Imports SeleniumBase
Imports OpenQA.Selenium
Imports OpenQA.Selenium.Support.UI
Imports System.Text

Public Class NAAccess
    Private Property mobjWebDriver As DriverBase

    Public Sub New(ByVal driverBase As DriverBase)
        If driverBase Is Nothing Then
            Throw New Exception("Invalid driver!")
        End If
        mobjWebDriver = driverBase
    End Sub

    Public Sub ConnectToNA()
mobjWebDriver.GetWebDriver.Navigate.GoToUrl("https://10.113.12.7/devicelist.query.execute.do?d
gfUpdate=true&deviceGroupID=1&fo_deviceGroupID>equals")

        If InStr(GetWebTitle.ToLower, "cannot display") Then
            Throw New Exception("Unable to connect to NA!")
        End If
    End Sub

    Public Sub ByPassCertificateMessage()
        If InStr(GetWebTitle.ToLower, "error") Then
```



```
mobjWebDriver.GetWebDriver.Navigate.GoToUrl("javascript:document.getElementById('overridelink')
).click()")
    End If
End Sub

' LOGIN
Public Sub TryToLoginNA(username As String, password As String)
    LoginNA(username, password)
End Sub

Private Sub LoginNA(username As String, password As String)
    'Este objeto nos permite ejecutar codigo javascript a traves del webdriver
    Dim objJavaScriptExecutor As IJavaScriptExecutor =
CType(Me.mobjWebDriver.GetWebDriver, IJavaScriptExecutor)

    Dim userElement As IWebElement =
mobjWebDriver.GetWebElementWaitUntilExists(By.Name("j_username"))
    objJavaScriptExecutor.ExecuteScript("arguments[0].value='" & username & "';",
userElement)

    Dim passwdElement As IWebElement =
mobjWebDriver.GetWebElementWaitUntilExists(By.Name("j_password"))
    objJavaScriptExecutor.ExecuteScript("arguments[0].value='" & password & "';",
passwdElement)

    Dim loginForm As IWebElement =
mobjWebDriver.GetWebElementWaitUntilExists(By.Name("LoginForm"))
    loginForm.Submit()
End Sub

Public Function getRoutersInfo() As String

    Try
        Dim datosRaw As String = mobjWebDriver.GetWebDriver.PageSource
        Return datosRaw
    Catch ex As Exception
        Return ""
    End Try
End Function

Private Function GetWebTitle()
    Return mobjWebDriver.GetWebDriver.Title
End Function

Public Sub Dispose()
    mobjWebDriver.Dispose()
    mobjWebDriver = Nothing
End Sub

End Class
```



Anexo G

Código Validador.vb

```
Imports Microsoft.Office.Interop
Imports DatabaseAbstractionLayer
Imports Renci.SshNet
Imports SocketConnection
Imports System.Text
Imports System.Runtime.InteropServices

Public Class Form1

    Private mObjLog As Logging.Log

    Public Sub New()
        InitializeComponent()
    End Sub

    Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        Dim resultadoTecnico As Boolean
        Dim resultadoVista As Boolean
        Try
            mObjLog = New Logging.Log("Validador Técnico")
            Dim logsFolder As String = My.Application.Info.DirectoryPath +
"\LOGS\ValidadorTecnico - " & Now().ToString("dd-MM-yyyy-HH_mm") & ".log"
            mObjLog.DefaultFileLogWriter.Location = Logging.LogFileLocation.Custom
            mObjLog.DefaultFileLogWriter.CustomLocation = logsFolder

            mObjLog.DefaultFileLogWriter.Location =
Logging.LogFileLocation.ExecutableDirectory

            Catch ex As Exception
                Console.WriteLine(ex.ToString)
            Finally
            End Try

            mObjLog.WriteEntry("AUTOMATISMO CERTIFICACIÓN IS2")
            mObjLog.WriteEntry("FECHA EJECUCIÓN: " + Now().ToString("dd-MM-yyyy-
HH_mm").ToString)
            mObjLog.WriteEntry(vbLf)

            resultadoTecnico = readHostnamesAndPing("ip", "user", "password")

            resultadoVista = executeView()

            mObjLog.DefaultFileLogWriter.Flush()

            Me.Close()
            Me.Dispose()
        End Sub

        Private Function readHostnamesAndPing(ByVal hostname As String, ByVal username As
String, ByVal passwd As String) As Boolean
```



```
Dim objSqlConnection As SqlConnection = Nothing
Dim objCheckRouterAnalyzer As CheckRouterAnalyzer = Nothing
Dim objCheckRouterAnalyzer2 As CheckRouterAnalyzer = Nothing
Dim sshConnection As SSHConnection = New SocketConnection.SSHConnection("ip",
"user", "password", Nothing, True)
Dim cpe_data As DataTable = Nothing

Try
    mObjLog = New Logging.Log("Router connectivity checking")

    objSqlConnection = New SqlConnection("user", "password", "bdd", "ip")
    Dim query1 As String = "EXECUTE INSERT_INT0_IS2 "
    objSqlConnection.ExecuteNonQuery(query1)
    cpe_data = objSqlConnection.ExecuteNonQuery("SELECT * FROM
[orange_cdm_industrializado].[dbo].[IS2_IMPORTED_DATA_DEF] WHERE [Código RFS] IS NOT NULL AND
[Resultado Final] = 0 AND [Tratar RFS]=1 AND ([Acceso IP]=0 OR [Configuración DNS]=0 OR [Alta
TACACS]=0)")

    If cpe_data.Rows.Count < 1 Then
        Exit Function
    End If

    ' Check actions
    Dim commandResult As New StringBuilder()
    Dim objTelnetIPConnectionCommand As Command = New Command("/usr/bin/telnet", "",
"", "username:user;password:", "Error while connecting through telnet")
    Dim objTelnetConnectionCommand As Command = New Command("/usr/bin/telnet", "",
"", "username:user;password:", "Error while connecting through telnet")
    Dim objLoginUsernameCommand As Command = New Command("XXX", "", "", "password:",
"Incorrect username or does not exist")
    Dim objLoginSuccessCommand As Command = New Command("XXX", "", "", "connected",
"Incorrect password or does not exist")
    Dim objExitRouterCommand As Command = New Command("exit", "", "", "[ocat-
pym@m2gred1:~]#", "Error on ""exit"" router session")
    objCheckRouterAnalyzer = New CheckRouterAnalyzer(New Command()
{objTelnetConnectionCommand, objLoginUsernameCommand, objLoginSuccessCommand,
objExitRouterCommand}, sshConnection)
    objCheckRouterAnalyzer2 = New CheckRouterAnalyzer(New Command()
{objTelnetIPConnectionCommand, objLoginUsernameCommand, objLoginSuccessCommand,
objExitRouterCommand}, sshConnection)
    Dim isCommandRun As Boolean = False
    Dim isCommandRun2 As Boolean = False
    Dim isOnline As Boolean = False
    Dim didLogin As Boolean = False
    Dim didConnect As Boolean = False
    Dim commandsRunResult As String = ""
    Dim commandsRunResult2 As String = ""

    For Each row As DataRow In cpe_data.Rows
        Dim id As String = CStr("" & row("ID"))
        Dim host As String = CStr("" & row("Router_cliente"))
        Dim ip As String = CStr("" & row("IP_loopback"))
        Dim x As Integer
        x = InStr(ip, "/")
        If x <> 0 Then
            ip = Mid(ip, 1, x - 1)
            row("IP_loopback") = ip
        End If

        Dim consoleKeyInfo As ConsoleKeyInfo = New ConsoleKeyInfo(Convert.ToChar(3),
ConsoleKey.C, False, False, True)
        If Not host.Equals("") Then
            isCommandRun = False
            isOnline = False
            didLogin = False
            didConnect = False
        End If
    End For
End Try
```



```
        commandsRunResult = ""
        sshConnection.Disconnect()
        sshConnection.Connect()
        objTelnetIPConnectionCommand.SetCommandParameters(ip)
        objTelnetConnectionCommand.SetCommandParameters(host)
        objLoginSuccessCommand.SetExpectedCommandResponse(host)
        isCommandRun = objCheckRouterAnalyzer.RunCommand()
        commandsRunResult = objCheckRouterAnalyzer.Analyze()
        isOnline = objCheckRouterAnalyzer.GetIsRouterOnline(commandsRunResult)
        If isOnline = True Then
            didConnect = True
            didLogin = objCheckRouterAnalyzer.GetRouterDidLogin(commandsRunResult)
            If didLogin = True Then
                isOnline = objCheckRouterAnalyzer2.Analyze(commandsRunResult)
                didConnect = objCheckRouterAnalyzer2.GetRouterDidConnect(commandsRunResult2)
                didLogin = objCheckRouterAnalyzer2.GetRouterDidLogin(commandsRunResult2)
            End If
        End If

        mObjLog.WriteEntry(commandsRunResult)
        mObjLog.WriteEntry("")
        mObjLog.DefaultFileLogWriter.Flush()
        Dim query As String = "UPDATE [orange_cdm_industrializado].[dbo].[IS2_IMPORTED_DATA_DEF] SET [Configuración DNS] = '" &
isOnline & "', [Acceso IP]=''" & didConnect & "', [Alta TACACS]=''" & didLogin & "'" & " WHERE [ID]=''" & id & "'"
        objSqlConnection.ExecuteNonQuery(query)
    End If
Next

Catch ex As Exception
    Console.WriteLine(ex.ToString)
    Return False
Finally
    If Not objCheckRouterAnalyzer Is Nothing Then objCheckRouterAnalyzer.Dispose()
    mObjLog.DefaultFileLogWriter.Flush()
    If Not objSqlConnection Is Nothing Then objSqlConnection.Dispose()
    If Not sshConnection Is Nothing Then sshConnection.Dispose()
End Try

Return True
End Function

Private Function executeView() As Boolean
    Dim objSqlConnection As SQLServerDataConnection = Nothing
    Dim viewResult As DataTable
    Try
        objSqlConnection = New SQLServerDataConnection("sa", "Sqladmin*15",
"orange_cdm_industrializado", "10.148.85.84")

        mObjLog.WriteEntry(vbLf)
        mObjLog.WriteEntry("ACTUALIZANDO IS2_IMPORTED_DATA CON DATOS DE LA WEB")

        Dim query1 As String = "EXECUTE INSERT INTO IS2 "
        objSqlConnection.ExecuteNonQuery(query1)

        mObjLog.WriteEntry(vbLf)
        mObjLog.WriteEntry("EJECUTANDO VISTA")
        mObjLog.WriteEntry(vbLf)
```



```

viewResult = objSqlConnection.ExecuteNonQuery("SELECT * FROM
[orange_cdm_industrializado].[dbo].[CERTIFICADOR_IS2_FINAL_DEF]")
System.Threading.Thread.Sleep(120000)

For Each row As DataRow In viewResult.Rows
    Dim okTareasWom As Boolean = False
    Dim okNA As Boolean = False
    Dim okHPINV As Boolean = False
    Dim resultadoFinal As Boolean = False
    Dim iteracion As Integer = 0
    Dim id As Integer = CInt(row("IS2_ID"))

    If row("OK Tareas WOM") = 1 Then
        okTareasWom = True
    End If

    If row("Alta HP Inventory") = 1 Then
        okHPINV = True
    End If

    If row("Alta NA Inventory") = 1 Then
        okNA = True
    End If

    Dim query2 As String = "UPDATE
[orange_cdm_industrializado].[dbo].[IS2_IMPORTED_DATA_DEF] SET [Alta HP Inventory]='' &
okHPINV & "', [OK Tareas WOM]='' & okTareasWom & "', [Alta NA Inventory]= '' & okNA & "'
WHERE [ID]='' & id
objSqlConnection.ExecuteNonQuery(query2)

    Dim lineasIS2 As DataTable
    Dim lineaIS2paraActualizar As DataRow
    lineasIS2 = objSqlConnection.ExecuteNonQuery("SELECT * FROM
[orange_cdm_industrializado].[dbo].[IS2_IMPORTED_DATA_DEF] WHERE [ID]='' + id.ToString()

    If lineasIS2.Rows.Count < 1 Then
        mObjLog.WriteEntry("ERROR: No se encuentra campo con la ID " +
id.ToString)
    ElseIf lineasIS2.Rows.Count > 1 Then
        mObjLog.WriteEntry("ERROR: Existe más de un registro con la misma ID " +
id.ToString)
    End If
    lineaIS2paraActualizar = lineasIS2(0)

    If Not IsDBNull(lineaIS2paraActualizar("TIPO ACCIÓN")) Then

        If lineaIS2paraActualizar("Acceso IP") And
lineaIS2paraActualizar("Configuración DNS") And lineaIS2paraActualizar("Alta TACACS") And
okHPINV And okNA And okTareasWom And lineaIS2paraActualizar("TIPO ACCIÓN") <> "BAJA" Then
            resultadoFinal = True
        End If
        If Not lineaIS2paraActualizar("Acceso IP") And Not
lineaIS2paraActualizar("Configuración DNS") And Not lineaIS2paraActualizar("Alta TACACS") And
Not okHPINV And Not okNA And okTareasWom And lineaIS2paraActualizar("TIPO ACCIÓN") = "BAJA"
Then
            resultadoFinal = True
        End If

    End If

    iteracion = CInt(lineaIS2paraActualizar("Iteración de Validación")) + 1

```



DISEÑO E IMPLEMENTACIÓN DE UNA HERRAMIENTA DE VALIDACIÓN E INVENTARIADO DE ROUTERS

```
Dim query As String = "UPDATE
[orange_cdm_industrializado].[dbo].[IS2_IMPORTED_DATA_DEF] SET [Resultado Final]='' &
resultadoFinal & "', [Iteración de Validación]='' & iteracion & "', [Fecha última Validación
gestión equipo]=getdate() WHERE [ID]='' & id
objSqlConnection.ExecuteNonQuery(query)

query = "SELECT * FROM
[orange_cdm_industrializado].[dbo].[INVENTARIO_CPES_DEF] WHERE [IS2_ID]='' & id.ToString
If objSqlConnection.ExecuteNonQuery(query).Rows.Count = 0 Then
    insertarRouterInventarioCPE(row)
End If

If resultadoFinal Then
    If compruebaRFS(lineaIS2paraActualizar) Then
        If envioCorreoRFSOK(lineaIS2paraActualizar) Then
            mObjLog.WriteEntry("Actualizando RFS completa a OK y correo
enviado: " + lineaIS2paraActualizar("Código RFS").ToString)
            Dim queryActualiza As String = "UPDATE
[orange_cdm_industrializado].[dbo].[IS2_IMPORTED_DATA_DEF] SET [Tratar RFS]=0, [Correo Enviado
a JP]=1 WHERE [Código RFS]='' + lineaIS2paraActualizar("Código RFS").ToString + ""
            objSqlConnection.ExecuteNonQuery(queryActualiza)
        Else
            mObjLog.WriteEntry("Actualizando RFS completa a OK con ERROR en
el envío del correo:" + lineaIS2paraActualizar("Código RFS").ToString)
            Dim queryActualiza As String = "UPDATE
[orange_cdm_industrializado].[dbo].[IS2_IMPORTED_DATA_DEF] SET [Tratar RFS]=0, [Correo Enviado
a JP]=0 WHERE [Código RFS]='' + lineaIS2paraActualizar("Código RFS").ToString + ""
            objSqlConnection.ExecuteNonQuery(queryActualiza)
        End If
    End If
End If
Next
Catch ex As Exception
    mObjLog.WriteEntry("ERROR: " + ex.Message)
Finally
    If Not objSqlConnection Is Nothing Then objSqlConnection.Dispose()
    mObjLog.DefaultFileLogWriter.Flush()
End Try

End Function

Private Function insertarRouterInventarioCPE(filaVista As DataRow) As Boolean
    Dim objSqlConnection As SQLServerDataConnection = Nothing
    Dim objListadoCPES As models.ListadoCPES

    Try
        mObjLog.WriteEntry("Insertando Router en inventario. ID: " +
filaVista("IS2_ID").ToString)
        objSqlConnection = New SQLServerDataConnection("sa", "Sqladmin*15",
"orange_cdm_industrializado", "10.148.85.84")
        objListadoCPES = New models.ListadoCPES(filaVista)
        objListadoCPES.InsertToDB(objSqlConnection, "INVENTARIO_CPES_DEF")
        Catch ex As Exception
            mObjLog.WriteEntry("ERROR Insertando en Inventario ID " +
filaVista("IS2_ID").ToString + ":" + ex.Message)
        Finally
            If Not objSqlConnection Is Nothing Then objSqlConnection.Dispose()
            mObjLog.DefaultFileLogWriter.Flush()
        End Try
    End Function

Private Function compruebaRFS(filaIS2 As DataRow) As Boolean
    Dim objSqlConnection As SQLServerDataConnection = Nothing
    Dim cpe_data As DataTable = Nothing
```



```
Dim RFScompletaOK As Boolean = True
Dim codigo As String
codigo = "Sin Código RFS"

Try
    If filaIS2("Código RFS").ToString <> "" Then
        codigo = filaIS2("Código RFS").ToString
    End If
    mObjLog.WriteEntry("Comprobando RFS: " + codigo)

    If codigo <> "Sin Código RFS" Then
        objSqlConnection = New SqlConnection("sa", "Sqladmin*15",
"orange_cdm_industrializado", "10.148.85.84")
        cpe_data = objSqlConnection.ExecuteNonQuery("Select * FROM
[orange_cdm_industrializado].[dbo].[IS2_IMPORTED_DATA_DEF] WHERE [Código RFS]='" +
filaIS2("Código RFS").ToString + "'")

        For Each row As DataRow In cpe_data.Rows
            If Not row("Resultado Final") Then
                RFScompletaOK = False
                Exit For
            End If
        Next
    Else
        RFScompletaOK = False
    End If

    Return RFScompletaOK
Catch ex As Exception
    mObjLog.WriteEntry("ERROR Comprobando RFS " + codigo + ":" + ex.Message)
Finally
    If Not objSqlConnection Is Nothing Then objSqlConnection.Dispose()
    mObjLog.DefaultFileLogWriter.Flush()
End Try

End Function

Private Function envioCorreoRFSOK(filaIS2 As DataRow) As Boolean
    Dim htmlBody As String = "La RFS con Código de RFS = " + filaIS2("Código
RFS").ToString + " de WOM = " + filaIS2("WOM").ToString + " , acción = " + filaIS2("TIPO
ACCIÓN") + " del router con Hostname = " + filaIS2("Router_cliente") + " está gestionada y
validada."

    Dim m_OutLook As New Outlook.Application
    Dim codigo As String = "Sin Código RFS"
    Try
        If filaIS2("Código RFS").ToString <> "" Then
            codigo = filaIS2("Código RFS").ToString
        End If
        mObjLog.WriteEntry("Enviando Correo para RFS: " + codigo)
        Dim objMail As Outlook.MailItem
        m_OutLook = New Outlook.Application

        objMail = m_OutLook.CreateItem(Outlook.OlItemType.olMailItem)

        objMail.BodyFormat = Outlook.OlBodyFormat.olFormatHTML

        objMail.To = "andres.alberto.quintero.valero@everis.com"
        objMail.Subject = "prueba de envio de correo - RFS OK"
        objMail.HTMLBody = htmlBody

        objMail.Send()
        mObjLog.WriteEntry("Correo enviado correctamente para RFS " + codigo)
```



```
        Marshal.ReleaseComObject(objMail)
        Return True
    Catch ex As Exception
        MessageBox.Show(ex.Message)
        Return False
    Finally
        Marshal.ReleaseComObject(m_OutLook)
        mObjLog.DefaultFileLogWriter.Flush()
    End Try

End Function

End Class
```

Anexo H

Vista SQL CERT_IS2_FINAL_DEF

```
SELECT DISTINCT
    IS2.ID AS IS2_ID, GETDATE() AS [Fecha Registro], IS2.[Fecha
Última Modificación] AS [Fecha Ultima Modificacion], IS2.Red, IS2.[TIPO
ACCIÓN], IS2.[WOM ALTA],
    (CASE WHEN IS2.[Código RFS] IS NOT NULL AND IS2.[Código RFS]
<> '' THEN IS2.[Código RFS] ELSE '' END) AS [Código RFS],
    IS2.CLIENTE AS [Datosadministrativosdecliente.Nombre], (CASE
WHEN FIT.[DIRECCIÓN] IS NOT NULL AND
    FIT.[DIRECCIÓN] <> '' THEN FIT.[DIRECCIÓN] WHEN OE.SEDE IS
NOT NULL AND OE.SEDE <> '' THEN OE.SEDE ELSE '' END) AS DirecciónSede,
    (CASE WHEN IS2.Router_cliente IS NOT NULL THEN
IS2.Router_cliente ELSE '' END) AS Router_cliente, (CASE WHEN IS2.[Orden
Nemonico] IS NOT NULL
    THEN IS2.[Orden Nemonico] ELSE '' END) AS OrdenNemonico,
    (CASE WHEN IS2.Tipoderouter IS NOT NULL THEN IS2.Tipoderouter ELSE '' END) AS
Tipoderouter,
    (CASE WHEN IS2.[Concentrador] IS NOT NULL THEN
IS2.[Concentrador] ELSE '' END) AS Nombre_RouterConcentrador, (CASE WHEN
IS2.[Interfaz concentrador] IS NOT NULL
    THEN IS2.[Interfaz concentrador] ELSE '' END) AS [Interfaz
concentrador], (CASE WHEN IS2.IP_loopback IS NOT NULL THEN IS2.IP_loopback ELSE
'' END) AS IP_loopback,
    (CASE WHEN IS2.[Subred conexión] IS NOT NULL THEN IS2.[Subred
conexión] ELSE '' END) AS [Subred conexión], (CASE WHEN IS2.DireccionWAN_RC IS
NOT NULL
    THEN IS2.DireccionWAN_RC ELSE '' END) AS DireccionWAN_RC,
    (CASE WHEN IS2.DireccionWAN_CPE IS NOT NULL THEN IS2.DireccionWAN_CPE ELSE ''
END)
    AS DireccionWAN_CPE, (CASE WHEN IS2.RD IS NOT NULL AND IS2.RD
NOT LIKE 'INTER%' THEN IS2.RD WHEN IS2.RD LIKE 'INTER%' THEN '0' ELSE '' END)
AS RD,
    (CASE WHEN IS2.VRF IS NOT NULL THEN IS2.VRF ELSE '' END) AS
VRF, (CASE WHEN IS2.[Código Línea] IS NOT NULL THEN IS2.[Código Línea] ELSE ''
END)
    AS [Código Línea], (CASE WHEN IS2.[Tecnología] IS NOT NULL
THEN IS2.[Tecnología] ELSE '' END) AS Tecnología, (CASE WHEN IS2.[Ancho de
Banda] IS NOT NULL
```



```

THEN IS2.[Ancho de Banda] ELSE '' END) AS AnchodeBanda, (CASE
WHEN IS2.[Numero Administrativo] IS NOT NULL THEN IS2.[Numero Administrativo]
ELSE '' END)
AS NúmeroadministrativooRTB, (CASE WHEN IS2.IOS IS NOT NULL
THEN IS2.IOS ELSE '' END) AS IOS, (CASE WHEN IS2.QoS IS NOT NULL THEN IS2.QoS
ELSE '' END)
AS QoS, (CASE WHEN IS2.SerialRouter IS NOT NULL THEN
IS2.SerialRouter ELSE '' END) AS SerialRouter, (CASE WHEN IS2.[Direccionamiento
Privado(sede)] IS NOT NULL
AND IS2.[Direccionamiento Privado(sede)] <> '' THEN
IS2.[Direccionamiento Privado(sede)] ELSE
(SELECT TOP (1) Sede
FROM dbo.WOM_IMPORTED_DATA AS W
WHERE (Proyecto = CAST(IS2.WOM AS NVARCHAR)) AND (Sede
IS NOT NULL)) END) AS [Direccionamiento Privado(sede)],
(CASE WHEN IS2.[Direccionamiento Publicas] IS NOT NULL THEN
IS2.[Direccionamiento Publicas] ELSE '' END) AS [Direccionamiento Publicas],
(SELECT TOP (1) [Persona Contacto Cliente]
FROM dbo.GESCIR_IMPORTED_DATA AS G
WHERE ([Código WOM] = CAST(IS2.WOM AS NVARCHAR) OR
[Código WOM] = CAST(OE.[CODIGO
PROYECTO] AS nvarchar) OR
[Código WOM] = OE.[COD_Tmp Proyecto]
OR
[Código WOM] = CAST(FIT.Id AS
NVARCHAR)) AND ([Persona Contacto Cliente] IS NOT NULL)) AS
Nombrecontactocliente,
(SELECT TOP (1) [Teléfono Contacto Cliente]
FROM dbo.GESCIR_IMPORTED_DATA AS G
WHERE ([Código WOM] = CAST(IS2.WOM AS NVARCHAR) OR
[Código WOM] = CAST(OE.[CODIGO
PROYECTO] AS nvarchar) OR
[Código WOM] = OE.[COD_Tmp Proyecto]
OR
[Código WOM] = CAST(FIT.Id AS
NVARCHAR)) AND ([Teléfono Contacto Cliente] IS NOT NULL)) AS
Teléfonocontactocliente, (CASE WHEN IS2.JP IS NOT NULL
THEN IS2.JP ELSE '' END) AS JPC,
(SELECT TOP (1) [Cod. Cli.]
FROM dbo.WOM_IMPORTED_DATA AS W
WHERE (Proyecto = CAST(IS2.WOM AS NVARCHAR)) AND
([Cod. Cli.] IS NOT NULL)) AS Códigodelcliente, IS2.WOM, (CASE WHEN IS2.[Tipo
Servicio] IS NOT NULL
THEN IS2.[Tipo Servicio] ELSE '' END) AS Servicio, (CASE WHEN
IS2.AccUsuari IS NOT NULL THEN IS2.AccUsuari ELSE '' END) AS AccUsuari,
(CASE WHEN IS2.AccDomini IS NOT NULL THEN IS2.AccDomini ELSE
'' END) AS AccDomini, (CASE WHEN
(SELECT COUNT(W.PROYECTO)
FROM WOM_IMPORTED_DATA AS W LEFT OUTER JOIN
WOM_TAREAS AS T ON W.TAREA = T
.[WOM_TAREA_DESCRIPCION]
WHERE T.[WOM_TAREA_FACTURABLE] = 'NO' AND W.PROYECTO
= IS2.WOM AND W.[Fecha Fin] IS NULL) = 0 THEN 1 ELSE 0 END) AS [OK Tareas WOM],

```



```
                (CASE WHEN HPINV.HOSTNAME IS NOT NULL AND HPINV.HOSTNAME <>
'' THEN 1 ELSE 0 END) AS [Alta HP Inventory], (CASE WHEN NA.HOSTNAME IS NOT
NULL AND
                NA.HOSTNAME <> '' THEN 1 ELSE 0 END) AS [Alta NA Inventory]
FROM      dbo.IS2_IMPORTED_DATA_DEF AS IS2 LEFT OUTER JOIN
                dbo.FIT_PROYECTOS AS FIT ON IS2.WOM = FIT.Wom LEFT OUTER JOIN
                dbo.ORDER_ENTRY AS OE ON IS2.WOM = OE.[CODIGO PROYECTO] LEFT
OUTER JOIN
                dbo.HPINV_IMPORTED_DATA AS HPINV ON IS2.Router_cliente =
HPINV.HOSTNAME LEFT OUTER JOIN
                dbo.NA_IMPORTED_DATA AS NA ON IS2.Router_cliente =
NA.HOSTNAME
WHERE (IS2.[Tratar RFS] = 1) OR
                (IS2.[Resultado Final] = 0)
```