



TECNOLÓGICO NACIONAL DE MÉXICO
INSTITUTO TECNOLÓGICO DE APIZACO
DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN

“DESARROLLO DE UN HEAD END PARA LA
GESTIÓN DE LA INFORMACIÓN DE UNA
RED INTELIGENTE”

TESIS

QUE PARA OBTENER EL GRADO DE:

MAESTRO EN SISTEMAS COMPUTACIONALES

PRESENTA:

LIC. AGUSTÍN SÁNCHEZ ATONAL

COMITÉ TUTORIAL:

DIRECTOR MC.JOSÉ JUAN HERNÁNDEZ MORA.

CO-DIRECTOR: M.C MARÍA GUADALUPE MEDINA BARRERA.

TUTOR: M.C. MARÍA JANAÍ SÁNCHEZ HERNANDEZ

REVISOR: M.C HIGINIO NAVA BAUTISTA

APIZACO, TLAXCALA

OCTUBRE 2015



TECNOLÓGICO NACIONAL DE MÉXICO
Instituto Tecnológico de Apizaco

"2015, Año del Generalísimo José María Morelos y Pavón"



AUTORIZACION DE IMPRESIÓN



TECNOLÓGICO NACIONAL DE MÉXICO
Instituto Tecnológico de Apizaco

"2015, Año del Generalísimo José María Morelos y Pavón"

Apizaco, Tlax., 24 de Septiembre de 2015

No. de Oficio: DEPI/286/15

ASUNTO: Se Autoriza Impresión de Tesis de Grado.

LIC. AGUSTIN SANCHEZ ATONAL
CANDIDATO AL GRADO DE MAESTRO
EN SISTEMAS COMPUTACIONALES
No. de Control: **M07370532**
P R E S E N T E.

Por este medio me permito informar a usted, que por aprobación de la Comisión Revisora asignada para valorar el trabajo, mediante la Opción: **I Tesis de Grado por Proyecto de Investigación**, de la **Maestría en Sistemas Computacionales**, que presenta con el tema: "**DESARROLLO DE UN HEAD END PARA LA GESTION DE LA INFORMACION DE UNA RED INTELIGENTE** " y conforme a lo establecido en el Procedimiento para la Obtención del Grado de Maestría en el Instituto Tecnológico, la División de Estudios de Posgrado e Investigación a mi cargo le emite la:

AUTORIZACIÓN DE IMPRESIÓN

Debiendo entregar un ejemplar del mismo debidamente encuadernado y seis copias en CD en formato PDF, para presentar su Acto de Recepción Profesional a la brevedad.

Sin otro particular por el momento, le envío un cordial saludo.

ATENTAMENTE

PENSAR PARA SERVIR, SERVIR PARA TRIUNFAR®

M.A.D. MA. A. ACELA DAVILA JIMENEZ
JEFA DE LA DIVISIÓN DE ESTUDIOS DE
POSGRADO E INVESTIGACIÓN.



Secretaría de Educación Pública
Instituto Tecnológico de Apizaco
División de Estudios de Posgrado
e Investigación

MAADJ/mebr.

C.p. Expediente.



Carretera Apizaco-Tzompantepec, esquina con Av. Instituto Tecnológico S/N
Conurbado Apizaco-Tzompantepec, Tlaxcala, Mex.
C.P. 90300 Apizaco, Tlaxcala. Tels. 01 241 41 7 20 10, Commur. 101 Ext. 146
www.itapizaco.edu.mx





TECNOLÓGICO NACIONAL DE MÉXICO
Instituto Tecnológico de Apizaco

"2015, Año del Generalísimo José María Morelos y Pavón"



AGRADECIMIENTOS

Agradezco a CONACYT por brindarme un apoyo económico durante el tiempo de estudio de la maestría y para las estancias realizadas en el estado de Querétaro.

A la empresa Softek Global eDesign que ahora forma parte del grupo EOS Technologies por permitirme realizar mis estancias, además de facilitar las instalaciones e información para el desarrollo del proyecto y la redacción de este documento.

A los profesores del Instituto Tecnológico de Apizaco por compartir un poco de su conocimiento para ayudar en mi formación académica; en especial a mi comité revisor por apoyarme en la redacción del presente.

A mis padres, hermana y familia por ayudarme en tiempos difíciles y enseñarme a ser perseverante para salir adelante.

A mis amigos y amigas; personas muy especiales, por apoyarme para salir adelante y alentarme a dar mi mejor esfuerzo en todo momento.

A el Lic. Carlos Domínguez Galván por enseñarme a enfrentar los retos sin pretextos y a disfrutar del trabajo.



TECNOLÓGICO NACIONAL DE MÉXICO
Instituto Tecnológico de Apizaco

"2015, Año del Generalísimo José María Morelos y Pavón"



DEDICATORIAS

Dedico estos momentos a mi recién nacido hijo por ser una razón más para salir adelante y a Jeaneth E. Quiriz Lopez por mostrarme otro lado de la vida que la inocencia y la rectitud son virtudes que jamás debemos de perder, por haberme ayudado tanto durante este tiempo, siendo mi mejor amiga y equipo de trabajo.

"The individual has always had to struggle to keep from being overwhelmed by the tribe. If you try it, you will be lonely often, and sometimes frightened. But no price is too high to pay for the privilege of owning yourself."

Friedrich Nietzsche



TECNOLÓGICO NACIONAL DE MÉXICO
Instituto Tecnológico de Apizaco

"2015, Año del Generalísimo José María Morelos y Pavón"



RESUMEN

El presente documento describe el desarrollo de un sistema informático de tipo Head End integrando la metodología ágil de SCRUM, programación extrema, programación por módulos y una serie de tecnologías las cuales fueron estudiadas e implementadas por el equipo desarrollador del cual se formó parte durante el periodo de estancias realizadas en la empresa Softek Global eDesign, la cual a su vez forma parte del grupo EOS Technologies; estas empresas facilitaron la información contenida en el presente documento para la descripción del proyecto desarrollado. Este proyecto fue financiado por CONACYT; institución que igualmente brindó apoyo económico a los alumnos de la Maestría en Sistemas Computacionales del ITA por medio de una beca durante el periodo de estudio.

En años recientes el uso ilícito del servicio de energía eléctrica brindado por la comisión federal de electricidad ha ido en aumento, la gente ha explotado fallas en medidores analógicos y digitales, así como en el sistema de cableado que se encuentra en funcionamiento en todo el país. Fallas en los sistemas de comunicación de la red de medidores digitales, así como la inyección de recursos que bien podrían ser aprovechados de otro modo, son parte de las razones por las cuales CFE; en busca de brindar un mejor servicio a sus usuarios, comienza un proyecto de modernización de la red eléctrica que ataque todos estos puntos brindando una funcionalidad confiable y decremente el uso de recursos tanto monetarios como humanos.

Dicho proyecto incluye el desarrollo de nuevos medidores lo cual creará una nueva estructura en la red eléctrica llamada Smart Grid o Red Inteligente. Este tipo de red se caracteriza por estar formada por dispositivos programados para funcionar de manera independiente, minimizando de este modo la interacción con recursos humanos, lo cual disminuye directamente la aplicación de estos recursos en campo, así como el gasto de recursos para desplazarlos hacia locaciones geográficas distantes o de cierto modo inaccesibles, que también representan un riesgo para el personal. A estos medidores se le unen dentro de la red inteligente un grupo de dispositivos que brindarán mayores facilidades al usuario como la desconexión remota de aparatos eléctricos. Todos los elementos que integran la nueva red de electricidad serán descritos en mayor detalle en este documento.

Por el otro extremo del sistema se encuentran las aplicaciones de software propias de la comisión llamadas SICOSS y SICOM, dentro de las cuales el personal de oficina se encarga de requerir la información a los medidores digitales que cuentan con la característica de comunicación remota, para poder realizar sus tareas cotidianas de facturación.

Para la integración de estas dos entidades se requiere el desarrollo de un Head End, el cual se encargará de brindar un puente de comunicación entre estas para que puedan interactuar.

Pandora Management System, como fue denominado este sistema cuenta con módulos para ofrecer esta funcionalidad. Por un lado se tiene el módulo de interacción con terminales portátiles para recabar información concentrada por el personal de campo al realizar las



TECNOLÓGICO NACIONAL DE MÉXICO
Instituto Tecnológico de Apizaco

"2015, Año del Generalísimo José María Morelos y Pavón"



instalaciones físicas desde los dispositivos y realizar los registros en las bases de datos correspondientes informando a los sistemas informáticos de los cambios realizados. El otro modulo es una interfaz web, la cual funciona como interfaz gráfica para que el personal de oficina pueda realizar las configuraciones iniciales para el funcionamiento de los dispositivos de la red inteligente asi como consulta de estadísticas de uso, errores y demás ocurrencias en esta.



TECNOLÓGICO NACIONAL DE MÉXICO
Instituto Tecnológico de Apizaco

"2015, Año del Generalísimo José María Morelos y Pavón"



ABSTRACT

This document describes the development of a Head End informatics system, this system integrates the SCRUM agile methodology, Extreme Programming, module programming, and a set of technologies which were studied and implemented by the developer team in which the author was part of during his professional practices in the enterprise called Softek Global eDesign, this enterprise is part of the EOS Technologies group. This project was funded by CONACYT, an institution that brings economical support for students in the ITA's Computing Systems Master.

In recent years illicit usage for the power energy service provided by the Federal Electricity Commission has increased. People are exploiting failures on analogic and digital meters, as well with the wiring system installed all over the country. Communication breakdowns over the digital meters grid, and wasting resources in tasks not worth of, are too some of the reasons why CFE; trying to bring a better user service, starts a modernization process for the power grid that covers all of this tasks giving a reliable functionality as well as decrease the usage of human and financial resources.

This project includes the development of new meters which will bring a new grid structure known as Smart Grid, this grid is integrated by some devices programmed to work independently, decreasing this way the interaction between the devices and human resources, which at the same time lowers the usage of this resources in field, and the expenses to send them to distant or inaccessible locations that might represent some risk for the workers.

This meters will work alongside with a group of other devices to bring new functionalities as remote disconnection for electrical devices. Every new device in the smart grid will be described later on this document.

On the other side of the overall system, there are some other software applications owned by the Commission called SICOSS and SICOM that are used by the office staff to ask for the information to the digital meters that can communicate remotely so they can use it for billing tasks.

To integrate this two entities the development of a Head End system is required, this system will work as a communication bridge so the Smart Grid and the Software systems can interact properly.

The so called Pandora Management System is conformed by modules that covers all this functions. At one side there is a module to collect the information registered by field staff on Hand Held devices and send it to a corresponding database sending a warning to the other software systems so they can know of the changes made. In the other side there is a web interface which works as a GUI for the office staff to do the initial configurations for the Smart Grid devices, as well as consult usage statistics, error statistics and some other matters that may have occurred on the system.



TECNOLÓGICO NACIONAL DE MÉXICO
Instituto Tecnológico de Apizaco

"2015, Año del Generalísimo José María Moreles y Pavón"



TABLA DE CONTENIDO

RESUMEN.....	I
ABSTRACT.....	III
TABLA DE CONTENIDO.....	IV
ÍNDICE.....	V
ÍNDICE DE TABLAS.....	VIII
ÍNDICE DE ILUSTRACIONES.....	IX

ÍNDICE

CAPITULO I INTRODUCCIÓN.....	1
DESCRIPCIÓN DEL PROBLEMA.....	2
PLANTEAMIENTO DEL PROBLEMA.....	6
JUSTIFICACIÓN.....	7
OBJETIVOS.....	9
OBJETIVO GENERAL.....	9
OBJETIVOS ESPECÍFICOS.....	9
ALCANCE Y LIMITACIONES.....	9
PREGUNTA DE INVESTIGACIÓN.....	9
DESCRIPCIÓN DEL CONTENIDO DEL DOCUMENTO.....	10
ESTADO DEL ARTE.....	10
CAPITULO II FUNDAMENTOS.....	25
REDES Y DISPOSITIVOS INTELIGENTES.....	26
ACCESO A ARCHIVOS XML.....	28
METODOLOGÍAS Y TÉCNICAS DE PROGRAMACIÓN.....	29



TECNOLÓGICO NACIONAL DE MÉXICO
Instituto Tecnológico de Apizaco

"2015, Año del Generalísimo José María Moreles y Pavón"



JAVA SERVER FACES Y PRIME FACES	39
SERVICIOS DE WINDOWS	41
JSP Y SERVLETS.....	42
CAPITULO III METODOLOGÍA.....	44
METODOLOGÍA PLANTEADA	45
IDENTIFICACIÓN DEL PROBLEMA.....	45
ARQUITECTURA DEL SISTEMA.....	48
REQUERIMIENTOS DEL SISTEMA	50
SELECCIÓN DE LOS LENGUAJES DE PROGRAMACIÓN.	57
SELECCIÓN DE LA BASE DE DATOS.....	58
DISEÑO DE LA BASE DE DATOS.....	64
DIAGRAMAS DE CASOS DE USO.....	69
DIAGRAMAS DE PAQUETES Y CLASES.....	76
DIAGRAMAS DE ESTADOS.....	93
HEAD END – PANDORA MANAGEMENT SYSTEM.....	95
MÓDULO WEB PANDORA WEB GUI.	96
HEAD END DAEMON.....	98
CAPITULO IV PRUEBAS Y RESULTADOS.....	100
RESULTADOS.....	101
PRUEBAS.....	121
CAPITULO V CONCLUSIONES Y TRABAJOS FUTUROS.....	125
CONCLUSIONES	126



TECNOLÓGICO NACIONAL DE MÉXICO
Instituto Tecnológico de Apizaco

"2015, Año del Generalísimo José María Moreles y Pavón"



TRABAJOS FUTUROS	129
REFERENCIAS BIBLIOGRÁFICAS.....	130
ANEXOS.....	133
A. CONCEPTOS Y ACRÓNIMOS.....	134
B. DOCUMENTO DE BACKLOG DE SPRINTS.....	137
C. CONSTANCIA DE SATISFACCIÓN DEL CLIENTE.....	145
D. CONSTANCIA DE LIBERACIÓN DE ESTANCIAS.....	147
E. PRODUCCIÓN ACADÉMICA.....	149



TECNOLÓGICO NACIONAL DE MÉXICO
Instituto Tecnológico de Apizaco

"2015, Año del Generalísimo José María Moreles y Pavón"



ÍNDICE DE TABLAS

Tabla 1.1 Extracto de comparación entre la red actual y una red inteligente Jixuan Zheng, (2013).....	10
Tabla 1.2 Arquitectura del proceso de medición para un medidor convencional. Jixuan Zheng, (2013).....	11
Tabla 1.3 Arquitectura para el proceso de medición con un medidor inteligente. Jixuan Zheng, L.L (2013).....	11
Tabla 1.4 Estructura de un equipo de SCRUM. (Yong, 2011).....	20
Tabla 1.5 Alternativas de lenguaje para Scrums que no son desarrolladores de software. Brent Barton (2009).....	21
Tabla 2.1 Comparación de la red actual contra la red inteligente. (Momoh, 2012) ...	26
Tabla 2.2 Habilidades interpersonales necesarias para la gestión de proyectos. (Institute, 2009)	30
Tabla 2.3 Comparativa de miedos y derechos de clientes y desarrolladores.....	35
Tabla 3.1 Características de las distintas versiones de Oracle (Precios de 2015)...	58
Tabla 3.2 Características de las distintas versiones de MySQL (precios de 2015)...	59
Tabla 3.3 My SQL versus ORACLE características / Funcionalidad.....	60
Tabla 3.4 Diseño conceptual de la tabla GMM	66
Tabla 3.5 Diseño conceptual de la tabla MM	66
Tabla 3.6 Diseño conceptual de la tabla MRE	67
Tabla 3.7 Diseño conceptual de la tabla PANDORA USERS.....	67
Tabla 3.8 Diseño conceptual de la tabla USER.....	68
Tabla 4.1 Registro de pruebas de funcionalidad y desempeño	122



TECNOLÓGICO NACIONAL DE MÉXICO
Instituto Tecnológico de Apizaco

"2015, Año del Generalísimo José María Moreles y Pavón"



ÍNDICE DE ILUSTRACIONES

Ilustración 1.1 Estructura general de la distribución geográfica de la red inteligente. . .	5
Ilustración 1.2 Scrum adaptado para el proceso de desarrollo de Software ágil. (Yong, 2011)	19
Ilustración 1.3 Aspectos de aseguramiento de la calidad en un ambiente de desarrollo con Scrum (Khalane & Tanner, 2013).....	22
Ilustración 1.4 Iteraciones aisladas. Jeff Sutherland (2005).....	23
Ilustración 1.5 Iteraciones traslapadas. Jeff Sutherland (2005).....	23
Ilustración 1.6 Todas a la vez. Jeff Sutherland (2005).....	24
Ilustración 2.1 Accesando al arbol DOM.....	28
Ilustración 2.2 Practicas de SCRUM. (Rubin, 2013).....	31
Ilustración 2.3 Ciclo de la programación extrema. (Chromatic, 2013).....	33
Ilustración 2.4 Arquitectura multicapa en Java. (Layka, 2014).....	37
Ilustración 2.5 Flujo de peticiones dentro de un JSP. (Brown, Dalto, & Jepp., 2011).	38
Ilustración 2.6 Arquitectura de un JSP. (Brown, Dalto, & Jepp., 2011)	39
Ilustración 2.7 Algunos de los componentes ofrecidos por PrimeFaces.(Hlvats, 2013)	40
Ilustración 2.8 Administrador de Servicios de Windows.	41
Ilustración 2.9 Modelo de Servlet. (Patzner, 2002).....	42
Ilustración 2.10 Modelo 1 de JSP.(Patzner, 2002).....	43
Ilustración 2.11 Modelo JSP 2. (Patzner, 2002).....	43
Ilustración 3.1 Diagrama a bloques de la interacción de los diferentes módulos del sistema.	46
Ilustración 3.2 Diagrama de bloques de la metodología vista de manera general.	47
Ilustración 3.3 Patrón de arquitectura de múltiples clientes – Un servidor para PANDORA MANAGEMENT SYSTEM.....	48
Ilustración 3.4 Diseño de arquitectura múltiples clientes – servidor con descripción de elementos.....	49
Ilustración 3.5 Diagrama a bloques de funcionalidad generalizada.	56

Ilustración 3.6 Diagrama Entidad - Relación Utilizado para el diseño de la base de datos.....	65
Ilustración 3.7 Diseño de la base de datos.....	68
Ilustración 3.8 Diagrama de caso de uso para la funcionalidad generalizada del sistema.....	70
Ilustración 3.9 Diagrama de funcionalidad general para el módulo Web.....	70
Ilustración 3.10 Diagrama de funcionalidad general para el módulo de terminales portátiles.....	71
Ilustración 3.11 Proceso para alta de servicio.....	73
Ilustración 3.12 Sección de cambios en el menú principal del Head End.....	74
Ilustración 3.13 Diagrama de caso de uso para edición de dispositivo.....	75
Ilustración 3.14 Diagrama de Paquetes.....	76
Ilustración 3.15 Diagrama Clase appBean.....	77
Ilustración 3.16 Diagrama Clase gmmBean.....	77
Ilustración 3.17 Diagrama Clase LogInBean.....	78
Ilustración 3.18 Diagrama Clase mmBean.....	78
Ilustración 3.19 Diagrama Clase menuBean.....	79
Ilustración 3.20 Diagrama Clase mreBean.....	79
Ilustración 3.21 Diagrama Clase userBean.....	80
Ilustración 3.22 Diagrama Clase varConfBean.....	81
Ilustración 3.23 Diagrama Clase writesXML.....	82
Ilustración 3.24 Diagrama clases gmmDao.....	83
Ilustración 3.25 Diagrama Clases gmmDaoImpl.....	83
Ilustración 3.26 Diagrama clase mmDao.....	83
Ilustración 3.27 Diagrama clase gmmDaoImpl.....	84
Ilustración 3.28 Diagrama clase mreDao.....	84
Ilustración 3.29 Diagrama clase mreDaoImpl.....	85
Ilustración 3.30 Diagrama clase varcConf.....	85
Ilustración 3.31 Diagrama clase varConfImpl.....	85
Ilustración 3.32 Diagrama clase repuDao.....	86



TECNOLÓGICO NACIONAL DE MÉXICO
Instituto Tecnológico de Apizaco

"2015, Año del Generalísimo José María Morelos y Pavón"



Ilustración 3.33 Diagrama clase rpuDaoImpl.	86
Ilustración 3.34 Diagrama de clase userDao.	86
Ilustración 3.35 Diagrama de clase userDaoImpl.	87
Ilustración 3.36 Diagrama clase Gmm.	87
Ilustración 3.37 Diagrama clase GmmlD.	87
Ilustración 3.38 Diagrama de clase Medidor Modular.	88
Ilustración 3.39 Diagrama de clase Medidor Modular ID.	88
Ilustración 3.40 Diagrama de clase de Módulo Remoto de Energía.	89
Ilustración 3.41 Diagrama de clase Módulo Remoto de Energía ID.	89
Ilustración 3.42 Diagrama de clase Lecturas.	90
Ilustración 3.43 Diagrama de clase Pandora Users.	90
Ilustración 3.44 Diagrama de clase Historial de Variables de Configuración.	91
Ilustración 3.45 Diagrama de clase User.	92
Ilustración 3.46 Diagrama de estados de la funcionalidad generalizada para el módulo de interfaz Web.	93
Ilustración 3.47 Diagrama de estados para la funcionalidad generalizada del módulo de conexión con terminales portátiles.	94
Ilustración 4.1 Captura de pantalla de LogIn.	101
Ilustración 4.2 Menú Principal del sistema.	102
Ilustración 4.3 Tareas para medidores modulares.	102
Ilustración 4.4 Interfaz para agregar un medidor.	103
Ilustración 4.5 Interfaz para edición de información de los medidores.	103
Ilustración 4.6 Ingreso de datos para la remoción de un medidor.	104
Ilustración 4.7 Lista de tareas para dispositivos de tipo MRE.	104
Ilustración 4.8 Ingreso de datos para agregar un MRE.	105
Ilustración 4.9 Ingresar para ingreso de datos de edición de MRE.	105
Ilustración 4.10 Interfaz para ingreso de datos para la remoción de MRE.	105
Ilustración 4.11 Lista de tareas para Gabinetes.	106
Ilustración 4.12 Interfaz para agregar un gabinete.	106
Ilustración 4.13 Lista de tareas para el interruptor de la puerta de un gabinete.	107



TECNOLÓGICO NACIONAL DE MÉXICO
Instituto Tecnológico de Apizaco

"2015, Año del Generalísimo José María Morelos y Pavón"



Ilustración 4.14 Ingreso de datos para habilitar el interruptor de puerta.....	107
Ilustración 4.15 Interfaz para deshabilitar el interruptor de puerta.....	107
Ilustración 4.16 Lista de tareas para la sección de Usuarios.....	108
Ilustración 4.17 Interfaz para ingreso de datos para agregar un nuevo usuario.....	108
Ilustración 4.18 Lista de tareas para servicios.....	109
Ilustración 4.19 Conexión de un nuevo servicio.....	109
Ilustración 4.20 Desconexión de servicio.....	110
Ilustración 4.21 Interfaz para ingreso de datos para conectar a un usuario.....	110
Ilustración 4.22 Interfaz para ingreso de datos para desconexión de usuarios.....	110
Ilustración 4.23 Eliminación de usuarios.....	110
Ilustración 4.24 Interfaz para eliminar usuarios.....	111
Ilustración 4.25 Lista de tareas para el menú de usuarios de Hand Held (Terminales Portátiles).....	111
Ilustración 4.26 Interfaz para introducir los datos de alta de usuario.....	112
Ilustración 4.27 Selección de tipo de rol de usuario.....	112
Ilustración 4.28 Lista de tareas para edición de información de usuario.....	113
Ilustración 4.29 Interfaz para la edición de nombre de usuario.....	113
Ilustración 4.30 Interfaz para la edición de contraseña de usuario.....	113
Ilustración 4.31 Interfaz para cambio de nombre de usuario.....	114
Ilustración 4.32 Interfaz para edición de rol de usuario.....	114
Ilustración 4.33 Eliminación de un usuario identificado por nombre de usuario.....	114
Ilustración 4.34 Selección de archivo para configuración.....	115
Ilustración 4.35 Cuadro de dialogo para seleccionar el origen del archivo.....	115
Ilustración 4.36 Lista de tareas para variables de configuración.....	116
Ilustración 4.37 Lista de tareas para variables de configuración de CCG.....	116
Ilustración 4.38 Lista de tareas para configurar tiempos de CCG.....	116
Ilustración 4.39 Interfaz para edición de tiempo de PLC.....	117
Ilustración 4.40 Lista de tareas para reintentos de CCG.....	117
Ilustración 4.41 Lista de tareas para configuración de periodos de CCG.....	117
Ilustración 4.42 Lista de tareas para MRE.....	118



TECNOLÓGICO NACIONAL DE MÉXICO
Instituto Tecnológico de Apizaco

"2015, Año del Generalísimo José María Morelos y Pavón"



Ilustración 4.43 Lista de tareas para configuración de tiempos de MRE 118

Ilustración 4.44 Lista de tareas para configuración de tiempos de pantalla de MRE.
..... 119

Ilustración 4.45 Lista de tareas para reportes de alarmas. 119

Ilustración 4.46 Pantalla de Historial de variables de configuración. 120

Ilustración 4.47 Interfaz de consulta de historial de variables de configuración. 120



TECNOLÓGICO NACIONAL DE MÉXICO
Instituto Tecnológico de Apizaco

"2015. Año del Generalísimo José-María Morelos y Pavón"



CAPITULO I INTRODUCCIÓN



TECNOLÓGICO NACIONAL DE MÉXICO
Instituto Tecnológico de Apizaco

"2015. Año del Generalísimo José-María Morelos y Pavón"



DESCRIPCIÓN DEL PROBLEMA

Softek Global eDesign una empresa ubicada en el estado de Querétaro que forma parte del grupo EOS Technologies, está dedicada a desarrollar ingeniería innovadora comienza un proyecto para la generación de una red inteligente diseñada para la Comisión Federal de Electricidad (CFE) encargada de brindar el servicio de energía eléctrica dentro de toda la república mexicana con la finalidad de automatizar procesos repetitivos que consumen una cantidad remarcable de recursos, al mismo tiempo que brinda agilidad a procesos de facturación, desconexión y re conexión de los servicios.

Una red inteligente o Smart Grid es una distribución geográfica de dispositivos inteligentes bajo una topología de Mesh Grid en la cual cada uno de los elementos que la integran puede comunicarse con cualquier otro dentro de la misma red estableciendo un canal de dos vías por el cual se transmite información hasta que esta es encaminada hacia un dispositivo colector de información o el elemento al cual iba dirigida dicha información. La comunicación de estos dispositivos se genera de manera inalámbrica vía Wi-Fi, Radio Frecuencia (RF), 3G, entre otras tecnologías.

De una manera más explícita, el proceso de comunicación inicia con un mensaje enviada desde fuera de la red hacia el dispositivo colector de información el cual basado en una tabla de direcciones es capaz de encaminar el paquete específicamente hasta el dispositivo solicitado. Una vez que el mensaje ha sido procesado el dispositivo puede generar una respuesta. La integración de la red en un área geográfica dependerá de la calidad de la conexión entre los dispositivos que la forman y el dispositivo colector; el cual trata de establecer comunicación con dispositivos cercanos integrándolos a una tabla de direcciones con la cual este podrá generar una lista de rutas para poder encaminar los mensajes. Por esta razón, si un dispositivo no está conectado directamente al dispositivo colector de información este podrá enviar su mensaje por una serie de dispositivos iguales a el que harán la función de repetidores reenviando la información hasta que esta alcance al colector.

Los dispositivos que integran a la red inteligente cuentan con capacidades para actuar de manera independiente una vez que son conectados y configurados. Estos pueden realizar sus tareas de manera periódica e incluso hasta diagnosticarse a sí mismos disminuyendo de este modo el uso de recursos. Las redes inteligentes pueden funcionar para prácticamente cualquier tipo de servicio, pero como se había

mencionado anteriormente; para este trabajo nos enfocaremos solamente en una red inteligente para la prestación del servicio eléctrico por parte de CFE.

El primer dispositivo que encontramos dentro de la red inteligente es el medidor modular (MM), un dispositivo inteligente el cual puede auto diagnosticarse para poder enviar un aviso oportuno de cualquier evento o falla. Este tipo de medidor realiza sus mediciones para el consumo del servicio recolectando información en tiempo real para la facturación dentro de la comisión, todo esto de manera automática, el hecho de que estos dispositivos puedan establecer una comunicación de dos vías facilita procesos como actualización de su firmware o solicitudes de información al momento sin colisiones de los paquetes de información que son enviados o recibidos.

El nombre de medidor modular viene del hecho de que estos dispositivos serán ubicados dentro de un gabinete al cual solo los trabajadores de CFE podrán tener acceso disminuyendo así el uso ilícito del servicio. Estos gabinetes son denominados gabinete modular de medidores(GMM), debido a que los medidores se encontraran conectados dentro del gabinete, estos tendrán dos salidas de comunicación; una salida por infra rojo (IR), y una salida por una antena RF.

El gabinete modular estará ubicado al costado del transformador o generador de energía por razones de seguridad, ya sea que este se encuentre en sótanos de edificios, en lo alto de los postes o como en algunos casos ubicado al nivel del suelo a un costado en la parte exterior del edificio. La razón primordial de la nueva ubicación del gabinete y la integración de los nuevos medidores dentro de él, es un esfuerzo para dificultar el acceso para personas con malas intenciones. Para el caso de las nuevas instalaciones eléctricas implementando los desarrollos de Softtek, si alguna persona infringe en el robo de energía, esta, técnicamente estará robando la energía de la línea de alguno de los usuarios registrados, resultando en un mayor consumo y por consecuente en un reporte de las irregularidades a la comisión decrementando de este modo la ocurrencia de este tipo de situaciones. En el caso de que un usuario intente invertir la posición del medidor y este mida erróneamente el consumo del servicio, el gabinete cuenta con sensores para saber si ha sido abierto; e incluso los mismos medidores tienen integrados sensores para dar un aviso al ser desconectados o girados de su posición original. Cabe mencionar que el mismo gabinete cuenta con guías de inserción para que los medidores solo puedan ser colocados de manera correcta.

Otra de las ventajas que ofrece este sistema es que el usuario podrá ver su consumo de energía estimado en una pantalla o MRE (Módulo Remoto de Energía). Este dispositivo entregado por la misma Comisión Federal de Electricidad de manera

gratuita al momento en que sea dado de alta un nuevo servicio, el MRE deberá ser ubicado dentro de cada hogar, y estará comunicado directamente al medidor correspondiente ya que de esta manera haciendo uso de la tecnología de comunicación PLC (Power Line Carrier) será capaz de recibir las lecturas que provienen de él. De este modo se ofrecerá al consumidor del servicio una manera fácil y cómoda para poder monitorear su consumo, además de que le sea posible ver un estimado de su consumo en pesos en tiempo real lo cual podrá ayudarlo a hacer una planeación de su consumo en el periodo corriente de facturación.

Dentro del mismo gabinete Modular también se encontrará una tarjeta de Control de Gabinete (TCG). Esta tarjeta de control es la encargada de recolectar la información de cada medidor ubicado dentro del gabinete; seleccionará la información que le es requerida (la cual será solicitada de manera periódica por los sistemas informáticos dentro de la comisión) como por ejemplo lecturas, perfiles de carga, etc. La TCG deberá de empaquetar la información y enviarla a través de la red inteligente de vuelta hacia las oficinas de CFE donde se verá reflejada dentro las bases de datos de su sistema informático o Utility y los sistemas de facturación.

El encargado de recolectar toda la información dentro de la red inteligente será un medidor característicamente de capacidades mayores y que también cuenta con una tarjeta LAN Ethernet para poder establecer comunicación vía el protocolo TCP/IP con el Head End, el cual a su vez se comunica con los sistemas informáticos dentro de las oficinas de CFE; este dispositivo es denominado coordinador. Este Edge Router o coordinador tendrá la función como su nombre lo dice, de hacer el papel de maestro debido a que es la conexión directa con CFE, entre sus características podemos encontrar mejores capacidades de procesamiento a nivel del micro procesador, además de la integración de la tarjeta Ethernet con la cual no cuentan los otros medidores dentro de la red inteligente; esta última tarjeta de red, deberá de ir conectada a una antena de radio frecuencia con mayor alcance y establecer conexión con la central de CFE simulando una conexión de red de área local cableada.

Como se había mencionado con anterioridad; estos dispositivos que forman la red inteligente, enviarán los datos generados hacia el Head End donde además de eso se realizan las tareas de asociación de dispositivos, medidores, tarjetas de control, y demás dispositivos integrados en la red hacia un usuario al que le es prestado el servicio con su correspondiente RPU, un identificador propio de la comisión para identificación de sus números de servicio. La comunicación entre estos dispositivos y la Utility de CFE para poder realizar todas las operaciones importantes para CFE como son lectura, altas bajas de servicio, reemplazo de dispositivos, etc.

Se realiza por la red de radio frecuencia que se genera con los Gabinetes Modulares, para después enviar la información a CFE mediante el colector.

Como puede observarse en la ilustración 1.1 sin importar el tipo de servicio (Monofásico, bifásico o trifásico) el usuario contará con un módulo remoto de energía y sus medidores según el caso del servicio; los cuales a su vez estarán ubicados dentro de un Gabinete Modular que estará colocado al lado del transformador en cualquiera que sea su ubicación (Suelo, poste, por debajo del suelo).

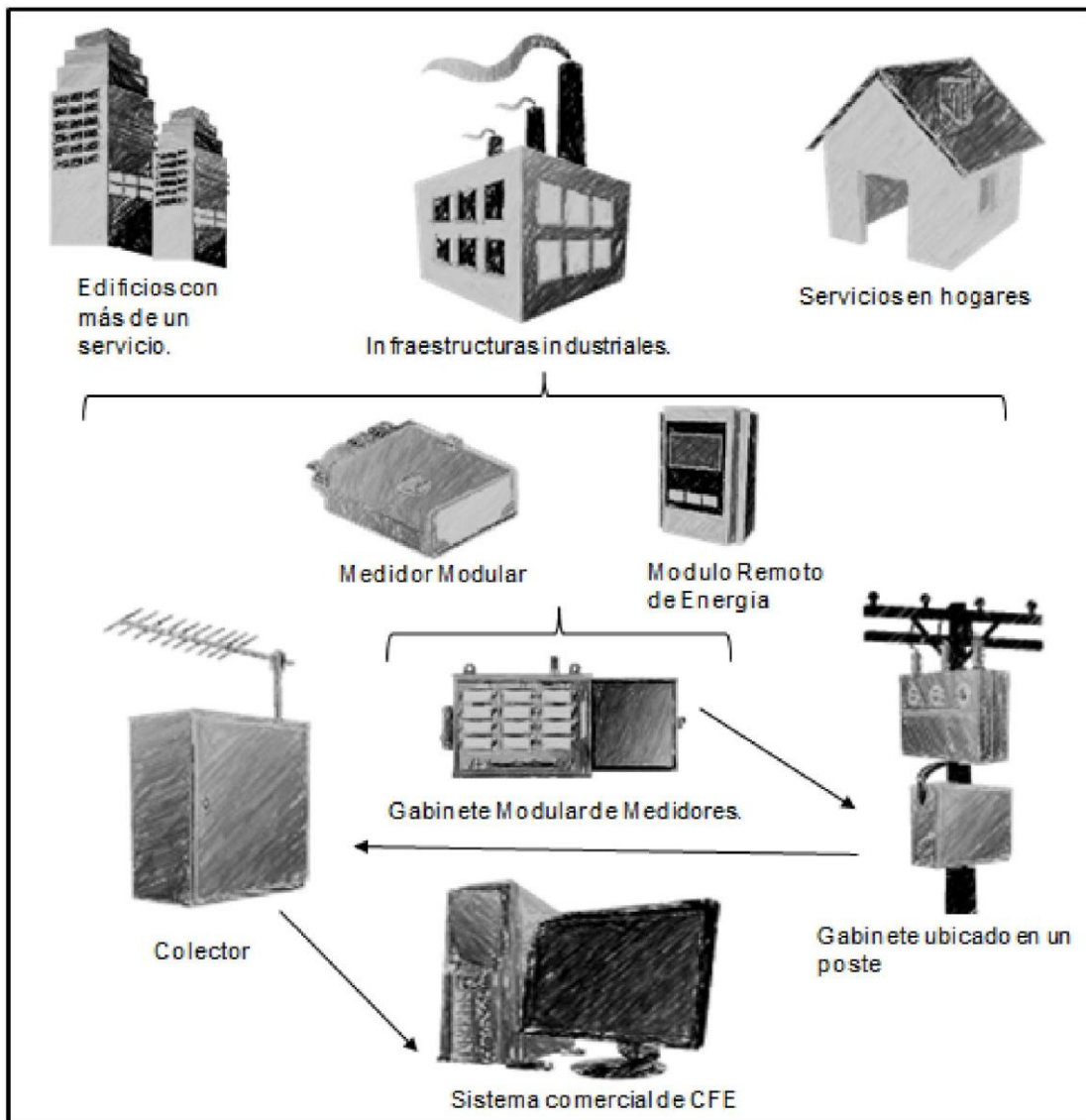


Ilustración 1.1 Estructura general de la distribución geográfica de la red inteligente.

Al comunicarse con el colector se emula una conexión Ethernet inalámbrica por la cual se transmitirán los datos hacia la comisión de electricidad directamente o bien a

través de una red de colectores hasta encontrar un dispositivo que cuente con una conexión de buena calidad y envíe finalmente de este modo la información generada por los medidores hacia los sistemas de facturación de CFE.

En la ilustración 1.1 se muestran los dispositivos que integran la red inteligente y como se distribuyen físicamente dentro de esta los servicios tendrán un medidor o tres de acuerdo a su conexión, mientras que se contará con un MRE para poder consultar un estimado de consumo en tiempo real.

Los gabinetes contenedores de los medidores se enlazarán al colector que sirve como una salida al exterior de la red estableciendo comunicación con el sistema informático de CFE. Para una mayor referencia de la distribución de la red y su funcionamiento (Lopez, 2015).

PLANTEAMIENTO DEL PROBLEMA.

Dentro del marco de desarrollo tecnológico descrito en la sección de introducción se encuentra la necesidad de desarrollar un sistema informático capaz de realizar la gestión de la información que fluye a través de la red inteligente desarrollada por Softek Global ya sea generada por los mismos dispositivos que la integran o información que es enviada desde la Utility.

Entre las necesidades para la programación del Head End se requiere el desarrollo de un software con la capacidad de sincronizarse con la Hand Held de CFE donde se registren todas las operaciones de campo para poder interpretar la información adquirida durante el día y que esta a su vez quede registrada en algún lugar como una base de datos.

Este software deberá de brindar una interfaz web para realizar ciertas operaciones como alta de usuarios, edición de usuarios, baja de usuarios, alta de dispositivos, edición de dispositivos, eliminar dispositivos.

Por otra parte, actualmente existe un software llamado NUBO que es el encargado de recibir los comandos de las operaciones que necesitan ser ejecutadas y deberá de enviarlas a las diferentes redes de medidores. El software a desarrollar se comunicará con NUBO para recibir esta información y enviarla a la red de gabinetes modulares desarrollada actualmente en Softek.(Lopez, 2015).

Dentro de la estructura de esta red fluyen ciertos paquetes de información requeridos ya sea periódicamente o manualmente, eventos desencadenados desde la interfaz web para ser solicitados a los dispositivos que conforman la red de medición.

La aplicación a desarrollar podrá ser programada para ejecutar las tareas mencionadas con anterioridad de manera periódica o en un instante determinado según su naturaleza, enviando las solicitudes generadas para las operaciones desde el Head End directamente al coordinador que funcionará como puerta de enlace de la red con el exterior de esta.

JUSTIFICACIÓN.

Actualmente los procesos de toma de lecturas implican una serie de gastos para la comisión federal de electricidad, el lectorista debe recibir una ruta establecida de lugares a los que debe dirigirse para atender solicitudes de servicios y manualmente tomar las lecturas medidor por medidor.

Dentro de un panorama de desarrollo e implementación de nuevas tecnologías para Smart Metering funcionando dentro de una red inteligente se incluye el desarrollo de un sistema capaz de gestionar el flujo de información dentro de esta para poder realizar las tareas para las que esta tecnología está siendo desarrollada y que a su vez sea capaz de poder coexistir con las tecnologías ya existentes como NUBO o Head End de EOS, sistemas de software que se encuentran funcionando actualmente y que a su vez interactúan con los sistemas de CFE.

Para una mejor comprensión de los requerimientos se comienza analizando algunos conceptos que resultan interesantes en el entendimiento de la funcionalidad de estas nuevas tecnologías.

Una Smart Grid es una red de dispositivos electrónicos inteligente, es decir una red integrada por ciertos dispositivos de medición en este caso del consumo de energía que son capaces de formarla y reintegrarla de manera automática. Estos dispositivos con características Plug & Play una vez energizados comenzarán a comunicarse en cierta frecuencia para poder integrar por primera vez los elementos de la red, y así establecer rutas de acuerdo a una característica llamada LQI (Link Quality Indicator) que como su nombre lo dice indicará cual es la ruta más óptima para cada nodo.

Una vez formada la red los nodos apuntarán a un colector que será su salida al mundo exterior, en este caso, la comunicación a CFE. Habiendo establecido este canal de comunicación lo que falta es la comunicación con la comisión Federal de Electricidad, para esto se deberá de desarrollar un software encargado de realizar dicha conexión y gestionarla.

Los dispositivos mencionados son whattorímetros encargados de la medición del consumo de energía, asociados a una pantalla que se entregará en cada casa de cada usuario para poder ver un estimado de su consumo así como un estimado en pesos de este. Los medidores estarán ubicados en un gabinete que contendrá a varios de ellos y contará con una tarjeta de control de gabinete que se podrá comunicar con otros gabinetes y colectores vía Radio Frecuencia.

Por otro lado podemos encontrar un sistema homólogo de nombre NUBO, que se encarga de tomar las peticiones para realizar tareas de SICOSS y SICOM de seccionarlas de acuerdo a los distribuidores y enviarlas al sistema correspondiente, en el caso del sistema en cuestión, el sistema utiliza a NUBO como único enlace hacia la comisión de electricidad, el cual deberá de seleccionar solo los medidores que correspondan a la red de Softek, empaquetando las tareas asignadas a estos y enviándolas por un WebService, que serán recibidas por el Head End Pandora y serán enviadas a través de la red inteligente hacia sus respectivos destinos para que puedan ser ejecutadas.

Para poder integrar todas las nuevas tecnologías descritas con anterioridad se debe de desarrollar una nueva tecnología implícita en un nuevo sistema de software que cumpla funciones de Head End, el cual es denominado "Pandora Management System" el cual estará dividido en tres partes de las cuales dos serán descritas en esta tesis. Estará formado por una Hand Held App, desarrollada de manera separada a las otras dos piezas de software que son las que serán tratadas aquí.

Estas dos piezas de software serán las encargadas de recolectar e interpretar la información proporcionada por la Hand Held para poder trabajarla y así enviarla para su almacenamiento a los lugares correctos y así se haga uso de ella por parte de la Comisión Federal de Electricidad.

Uno de los objetivos primordiales para el desarrollo de este software es generar una homogeneidad con los otros sistemas ya existentes y con los que deberá de tener comunicación continuamente de allí nos hemos basado para poder seleccionar el correcto lenguaje de programación y plantear la funcionalidad de los nuevos desarrollos.

El software a desarrollar deberá de ser capaz de intermediar esta comunicación, siendo esa la principal razón por qué el sistema es apoyado en su base de datos propia guardara las asociaciones de los nuevos dispositivos y cuando se realice el proceso de envío de tareas complementara las solicitudes con los datos de la asociación para la correcta gestión de los paquetes dentro de la red.

OBJETIVOS

OBJETIVO GENERAL

Desarrollar un sistema integral de tipo Head End que sea capaz de gestionar el flujo de información a través de la red inteligente de medidores de CFE, desarrollada por la empresa de ingeniería Softek Global eDesign. Este sistema estará encargado de comunicar las peticiones de tareas generadas desde la Utility de la comisión Federal de electricidad, hacia los medidores específicos integrados dentro de la red.

OBJETIVOS ESPECÍFICOS.

- Sincronización con terminales portátiles Hand Held.
- Interpretación de archivos XML.
- Generación de archivos XML para la red de medidores.
- Generación de archivos XML para la configuración de la terminal portátil.
- Generación de base de datos de PMS (Pandora Management System) en ORACLE.
- Conexión de los módulos de la aplicación programados en JAVA con la Base de datos.
- Funciones de inserción, edición y borrado de los módulos de la aplicación programados en JAVA.
- Generación de interfaz Web mediante un Servlet Java con JSP.

ALCANCE Y LIMITACIONES

La participación dentro del desarrollo dentro del proyecto general solo cubre el diseño y programación de las funciones más elementales dentro del Head End, dentro del tiempo de desarrollo se integraron compañeros del instituto tecnológico de Apizaco para complementar la programación de ciertas tareas. (Palafox, 2015).

PREGUNTA DE INVESTIGACIÓN.

¿Cómo desarrollar un Head End que sea capaz de manejar y asociar dispositivos para Smart Metering y comunicarlos con la Utility propia de la Comisión Federal de Electricidad utilizando metodologías ágiles para el desarrollo?

DESCRIPCIÓN DEL CONTENIDO DEL DOCUMENTO.

El presente documento está estructurado en cinco capítulos los cuales cubren en su totalidad al proceso para el desarrollo del proyecto comenzando con una breve introducción que describe los conceptos básicos para el entendimiento del funcionamiento de una red inteligente y como esta interactúa con el Head End y la importancia de este, el establecimiento de los objetivos de este trabajo al igual que el alcance y sus limitaciones; además de una breve reseña de trabajos relacionados a manera de estado del arte.

ESTADO DEL ARTE

(Jixuan Zheng, 2013) Se entiende por red eléctrica inteligente como aquella formada por la central de generación de la energía eléctrica; es decir, líneas para la transmisión de la electricidad, sistemas para la distribución de la energía y el cliente final o consumidor final.

Tabla 1.1 Extracto de comparación entre la red actual y una red inteligente Jixuan Zheng, (2013)

Red Actual	Red Inteligente
Electromecánica	Digital
Comunicación en solo un sentido	Comunicación de dos vías
Generación centralizada	Generación distribuida
Pocos sensores	Sensores por toda la red
Monitoreo manual	Auto monitoreo
Restauración manual	Auto reparación
Fallas y apagones	Adaptativo y segmentado

Dentro de cualquier red eléctrica inteligente uno de los dispositivos con mayor importancia es el medidor inteligente o Smart Meter, dispositivo que ofrece como principal ventaja un sistema de comunicación de dos vías, brindando de este modo automatización y una mejor distribución de la energía dentro de la red.

Este dispositivo es un medidor de energía avanzado el cual obtiene información del usuario generando la medición del consumo y al mismo tiempo que la complementa con datos de uso y dispositivos conectados para que esta realice sus tareas de facturación.

Toda esta información es medida en tiempo real y se guarda en un registro dentro del mismo dispositivo. La comunicación para el envío de dicha información hacia la central se realiza de manera bidireccional permitiendo a su vez poder conectarlo, desconectarlo, monitorearlo y controlarlo de manera remota.

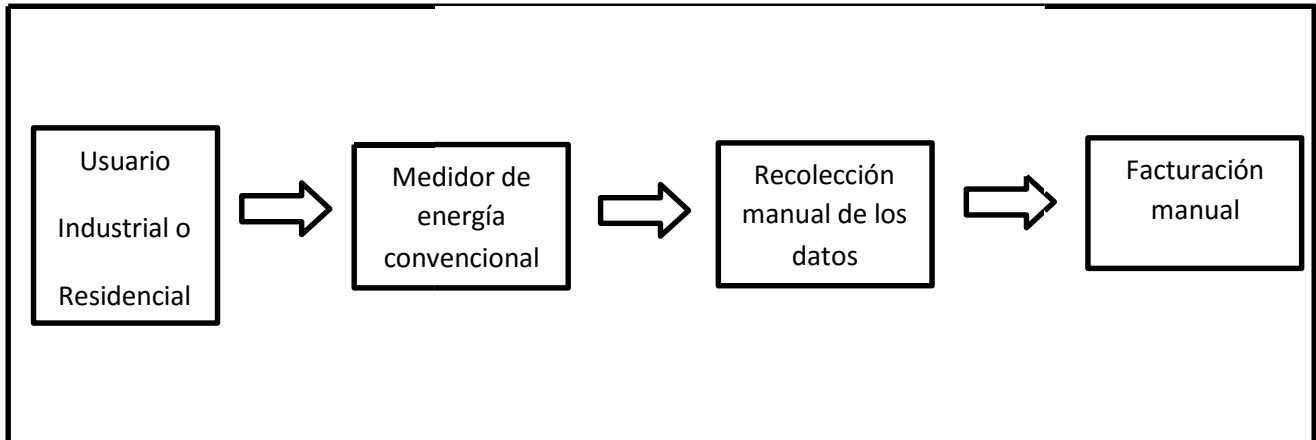


Ilustración 1.2 Arquitectura del proceso de medición para un medidor convencional. Jixuan Zheng, (2013)

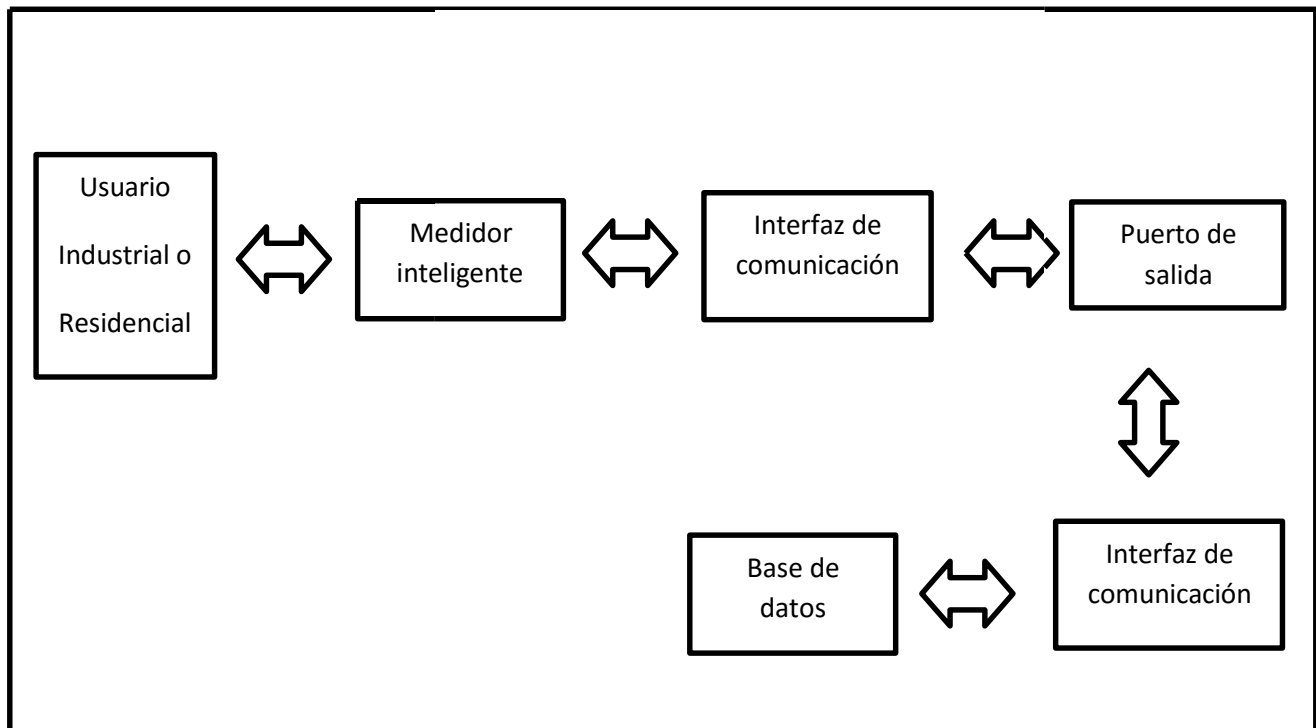


Ilustración 1.3 Arquitectura para el proceso de medición con un medidor inteligente. Jixuan Zheng, L.L. (2013)

Los medidores inteligentes comunes deben contar con por lo menos algunas de las siguientes características:

- Capacidad para comunicación de dos vías.
- Recolección de datos.
- Poder grabar los registros obtenidos.
- Controlar la carga.
- Funciones para ser programados.
- Facturación.

En cuanto al proceso de comunicación dentro de la arquitectura de la red inteligente encontramos que los medidores inteligentes recolectan los datos del servicio al que están relacionados, dentro de una red de área local o LAN, que tiene como punto de acceso un colector. Este proceso de comunicación puede ser realizado de manera periódica adaptándose a las necesidades del prestador de servicio. Una vez que el colector recibe la información de los medidores dentro de su red, este la envía hacia la Utility por una red WAN; haciendo uso de tecnologías PLC (Power Line Carrier) o RF (Radio Frequency).

Un panorama del desarrollo global es descrito en (Jixuan Zheng, 2013) donde se menciona que en Francia los mayores proveedores del servicio de energía eléctrica y gas son las empresas Electricité De France (EDF) y Electricité Réseau Distribution France (ERDF) las cuales buscan principalmente la implementación de redes inteligentes cuya contribución sea en áreas de facturación y servicio al cliente, conexión y desconexión remota, contabilidad, monitoreo y operación de la red.

Para Alemania la situación es diferente debido a que sus legislaciones son poco claras para el establecimiento de redes inteligentes, por lo cual tienen poco desarrollo y su implementación llega a generar costos extra para los usuarios; por lo cual solo clientes con altos consumos registrados son los que más implementan estas tecnologías.

Sin embargo en Holanda el desarrollo de las tecnologías inteligentes tiene gran auge, permitiendo a diversas empresas competir para prestar dicho servicio y a los usuarios escoger inclusive a que grado puede integrarlas, con medidores inteligentes con capacidades completas, limitadas o que inclusive sea necesario la toma de lecturas de forma manual; gracias a dos políticas establecidas en 2012.

Para el caso de Noruega, la aplicación de medidores automatizados tiene un gran desarrollo debido a que sus legislaciones permiten una libre competencia lo cual permite a los proveedores establecer diferentes tarifas.

(Simonov, 2014) En la actualidad la medición del uso del servicio eléctrico de una forma inteligente permite manejar a voluntad la dinámica para la producción y el consumo de energía mediante dispositivos inteligentes de medición propiedad de las

Utilities de energía que al integrarlas en una topología de red convierten a los dispositivos en fuentes de una gran variedad de información para análisis.

Una de las ventajas que brinda la medición en tiempo real que de este modo se permite que los medidores inteligentes es el poder de proveer una imagen de lo que está pasando en un preciso instante dentro del sistema de energía.

El artículo de (Simonov, 2014) presenta un enfoque orientado al método de manejo de tiempos. El cual presenta una problemática en cuanto a la producción de datos; el intercambio de datos de medición genera un flujo grande de datos lo cual implica un costoso tráfico de datos a nivel computacional y a nivel comunicación, lo cual podría en cierto momento saturar las redes de comunicación.

Haciendo una comparación de la producción de información dentro de las redes de medición de uso del servicio, bajo diferentes enfoques, se puede observar que los medidores inteligentes envían datos a un intervalo estimado de 15 minutos lo cual implica que este provee datos aproximadamente 96 veces por día, más sin embargo en un enfoque orientado a eventos es implícita la generación de datos aproximadamente cada 20 milisegundos a través de la red por cerca de 20 millones de nodos dentro de esta lo cual generaría un estimado de 86 Peta Bytes de información diariamente. Estos datos deberán de ser tomados en cuenta al momento del diseño de la red para poder realizar un estimado en los costos. Dentro de una red de medición inteligente la red avanzada de medición (AMI) deberá de permitir realizar interacciones con los dispositivos remotos.

Desde la perspectiva del cliente, los dispositivos ofrecen un valor agregado al proveer información para que este consciente de su consumo de energía al igual de poder planificar sus gastos para generar un ahorro.

Mientras que por el lado de la Utility le permitirá monitorear tendencias en el consumo de ciertas áreas para poder manejar de una mejor manera la carga que se brinda en estas e incluso una mejor planeación del posicionamiento geográfico de sus plantas generadoras.

En el artículo (Simonov, 2014) se presenta un método híbrido que combina los enfoques orientados al manejo de tiempo y de eventos en la medición eléctrica el cual tiene como objetivo proveer datos mayormente significativos, haciendo más eficiente de este modo el flujo de la información dentro de la red inteligente y las cargas de esta que son producidas por los dispositivos de metrología inteligentes. El enfoque en tiempo real presenta una fuente para la predicción de los niveles de energía esperados presentando las variaciones en la energía. Brindando además a

detalle información acerca de las dinámicas del flujo de energía mediante un monitoreo de los procesos en tiempo real, lo cual genera información importante en demasía para ser presentada a la Utility y que además nos brinda un panorama de las cargas de información que deberán de fluir a través de la red inteligente.

(Dong, Hongxing, Tianmiao, & Suibing, 2012) menciona que en orden para poder generar una red inteligente con una mayor eficiencia, seguridad, confiabilidad y sustentabilidad, es necesario integrar una herramienta de monitoreo y diagnóstico la cual es denominada WSN por sus siglas en inglés (Wireless Sensor Network); la cual busca la integración de herramientas de la informática, tecnología y comunicación con la red eléctrica. Estas herramientas pueden cubrir grandes áreas geográficas con un costo bajo aplicando los nuevos avances realizados en el campo de la radio frecuencia.

Las redes de energía eléctrica son una de las redes más grandes y complejas creadas y aun así han podido ser mejoradas con la implementación de tecnologías propias de las TIC. Como resultado de esta integración surgen las redes inteligentes que emplean diversas tecnologías de comunicación tales como líneas de energía (PLC), satélite, fibra óptica, redes de teléfono celular, redes Mesh y redes inalámbricas de rango corto dependiendo de su ubicación geográfica.

Las redes de sensores inalámbricas o WSN actualmente proveen herramientas que pueden ser integradas a una red inteligente como soluciones altamente confiables. Aunque estas presentan algunas problemáticas tales como tiempos de vida de baterías limitados y la baja capacidad para enviar grandes paquetes de datos. Estas problemáticas han sido atacadas con la implementación de redes WI-FI y una prometedora nueva técnica denominada Energy Harvesting o cosecha de energía. La interacción entre las redes inteligentes y las redes de sensores ha tenido recientes aplicaciones por ejemplo en Kentucky, en una subestación; fue implementada una WSN la cual utilizaba un protocolo de calidad de enlace el cual se implementa midiendo la calidad de la conexión para poder formar una ruta de comunicación viable entre los nodos que conforman la red.

Otra de las aplicaciones muestra una implementación de sensores multimedia y redes de actores dentro de la red inteligente. Lo cual demuestra las diversas aplicaciones de estas tecnologías no solamente para redes de energía.

(EROL-KANTARCI & MOUFTAH, 2012) Menciona que el formato de archivos XML (eXtensible Markup Lenguaje) tiene una gran versatilidad en cuanto a sus aplicaciones, al igual de una gran variedad de herramientas con las que se puede contar para manejar este formato de archivos.

Una de estas herramientas mencionadas en el artículo, es la llamada Dockbook una aplicación madura y bien probada. Dockbook es un lenguaje XML orientado a la producción de libros y el manejo de la producción y organización de contenido para enseñanzas, el cual cuenta a su vez con herramientas de código abierto y pueden ser adecuadas para ajustarse a cualquier necesidad.

(Téllez, 2009) El lenguaje XML ha ganado gran interés desde su introducción en 1988; a tal grado que se ha convertido en un estándar para el intercambio de una gran cantidad de datos a través de la red. La sintaxis de un lenguaje basado en XML es especificado haciendo uso de elementos y atributos en conjunto con algunas reglas para restringir su uso.

Existe una gran variedad de lenguajes de esquema para XML entre los cuales se encuentran:

- DTD.
- XML Schema.
- DSD2.

Un archivo XML se considera válido cuando se adapta de manera correcta a cualquiera de las sintaxis de cualquiera de estos esquemas. Otra de las adaptaciones más comunes del lenguaje XML es el XHTML una versión adaptada de HTML la cual es usada para servicios Web interactivos donde los usuarios interactúan mediante una interfaz con algún servidor.

En cuanto al desarrollo de documentos XML desde un lenguaje de programación de los denominados de propósito general, no se encuentra con ninguna ayuda específica para realizar esta tarea, por lo cual el programador puede optar por escoger de entre dos alternativas:

- Crear el documento como cadenas de texto.
- Crear el documento haciendo uso de estructuras DOM.

Los autores del artículo consideran como una meta propia el poder integrar XML dentro de Java para ayudar a realizar Web Services de manera más segura y rápida.

Debido a que el esquema de XML define una clase de documentos XML, se puede ver a estos esquemas como tipos. El objetivo principal del desarrollo descrito es la transformación de archivos XML, proceso mediante el cual se toman varios archivos XML para estandarizarlos en una salida de un solo archivo XML.

Dentro de las contribuciones descritas se encuentran:

- La generación de un tipo de datos que puede realizar operaciones de alto nivel definiendo transformaciones de XML en Java.
- Una técnica de análisis estático basada en una noción de grafos de resumen.
- Un algoritmo para evaluación simbólica de expresiones XPath en grafos de resumen.
- Un algoritmo para convertir esquemas DTD a grafos de resumen.

Dentro de este trabajo se escogió el lenguaje de programación Java debido a que ya es utilizado mayormente en el desarrollo de servicios Web, y permite la integración de XML con algunas otras funcionalidades tales como acceso a bases de datos o la Internet. Se comenzó el desarrollo descrito en el artículo con la implementación de XPath para obtención de los valores dentro de los documentos XML presentándolos como templates o fragmentos bien formados de XML que pueden contener espacios para que otros datos sean integrados.

DTD se considera un esquema o gramática establecida para una clase XML en la cual para cada elemento existen elementos hijo y atributos. XPath a su vez es un DSL para la ubicación de elementos, atributos y datos de caracteres o nodos en documentos XML. El sistema desarrollado en este trabajo (Téllez, 2009) provee de este modo un enfoque de alto nivel para la manipulación de datos XML en Java y un programa para la validación estática de documentos integrando las tecnologías mencionadas.

(Carl Binding, 2000) Menciona en su artículo que el incremento en el uso de Internet ha generado constantes avances tecnológicos, algunos de estos son generalmente páginas HTML interconectadas mediante links, que eventualmente fueron mejoradas con la implementación de Servlets Java los cuales son clases de Java que cuentan con una interfaz Servlet. Se ingresa a estos Servlets desde un servidor HTTP que solicita ejecutar los procesos del Servlet dentro del servidor y a su vez genera una respuesta representada como una instancia de una clase de Java.

JSP o Java Server Pages consiste en templates Mark-up dentro de los cuales se puede insertar fragmentos de código, y dentro de estos mismos elementos pueden ser integrados los Java Beans lo cual brinda un método para poder segmentar los aspectos de la interfaz en módulos JSP y los aspectos de acceso a datos en los Java Beans. XML es un metalenguaje que permite especificar lenguaje de marcas, su sintaxis se encuentra dentro del "Document Type Definition (DTD)" o definición de tipo de documento en español.

Dentro de este se pueden especificar elementos que representan entidades estructuradas identificadas por un símbolo de "menor que"; y puede contener

atributos que pueden contener valores por default y por características pueden ser incluidos de manera optativa. Estos elementos pueden contener otros elementos en una estructura jerárquica.

Los Java Beans son componentes de Java que brindan métodos de acceso para establecer valores y recibir valores de propiedades de ciertos datos. Desde el punto de vista de JSP los Java Beans forman una conexión entre XML y fuentes de datos.

(Adam Dukovich, 2008) Describe una librería para mapear documentos XML llamada Java Object Mapping (JOXM) funciona con persistencia automatizada y con consultas de objetos de Java en una base de datos de XML nativa. Brindando de este modo una alternativa para el mapeo objeto – relacional del mismo modo que Hibernate. En la mayoría de las estructuras de datos en la red se usa el formato XML, estructurado bajo arboles DOM (Document Object Model) por lo cual las aplicaciones para el mapeo de estos objetos hacen uso de ORM o mapeo objeto relacional. Usualmente se puede hacer uso de Hibernate para asegurar el almacenamiento persistente dentro de la base de datos sin la necesidad de incorporar desarrollo de base de datos en la aplicación.

JOXM fue diseñado para sustituir a ORM en aplicaciones de software aunque aplica ciertos conceptos propios del mismo, como la abstracción y simplicidad de la API sin necesidad de integrar actividades de desarrollo y administración de base de datos.

(Yong, 2011) En años recientes la implementación de la metodología ágil SCRUM ha ido ganando importancia dentro del área de desarrollo de software; permitiendo a los equipos desarrolladores un mejor desempeño, confiable y de calidad.

Los métodos tradicionales a diferencia de las metodologías ágiles son difíciles de controlar además de brindar una gran cantidad de dificultades como comprobar el cumplimiento de los requerimientos hasta fases avanzadas. Sin embargo las metodologías ágiles facilitan el proceso de cumplimiento de requerimientos presentando posibilidades de hacer entregas periódicas al dividir el desarrollo en fases más pequeñas mediante las cuales se entregan avances al cliente comprobando de este modo el cumplimiento de los alcances con calidad y facilitando la adaptación a cambios, adición de nuevas funcionalidades y mejoras a la eficiencia de la producción, además de mejorar continuamente el desempeño del equipo desarrollador.

SCRUM es un marco de trabajo que depende totalmente de la experiencia previa del equipo de desarrollo, delegándole completamente la responsabilidad de hacer un buen trabajo para las tareas de planeación, diseño, desarrollo y pruebas.

Dentro de la metodología se definen tres roles principales:

- El Scrum Master, que es el líder del equipo y basado en su experiencia se encarga de la asignación de tareas a los miembros del equipo, además de brindar la confianza y apoyo a este ante cualquier problemática que pueda surgir.
- El equipo, encargado de llevar a cabo las tareas específicas asignadas por el Scrum Master.
- El propietario del producto al que son presentados los resultados.

Para desempeñar todas las tareas se planean una serie de entregas a las que se les asigna una fecha final llamados Sprints; en estos, se hacen revisiones de los avances logrados al igual que de las problemáticas encontradas, lo que brinda las guías para el inicio del siguiente Sprint realizando los registros indicados en Sprint Backlogs (Documentación de registro de tareas desempeñadas por Sprint).

Como ejemplo de implementación el autor implementa la mencionada metodología en un sistema de manufactura de refacciones para automóviles seleccionando un Framework pre diseñado de Visual Studio 2010, el cual tras una capacitación del personal involucrado en el desarrollo fue adaptado y optimizado de acuerdo a sus necesidades.

(Mundra, Barclays, & Misra, 2013) Comenta que hoy en día la naturaleza de los desarrollos de software ha cambiado debido a que los requerimientos cambian de manera frecuente, haciendo de cierto modo un poco más difícil el llevarlos a cabo; gracias a su flexibilidad y adaptabilidad SCRUM surge como una alternativa completa para este tipo de proyectos.

Esta metodología trabaja segmentando grupos de trabajo en grupos de menor tamaño con la finalidad de una mejor interacción facilitando el alcanzar los objetivos propuestos para cada uno de ellos; estos grupos son llamados Scrum de Scrums (SoS por sus siglas en inglés) los cuales, según estudios arrojan mejores resultados que grupos relativamente grandes.

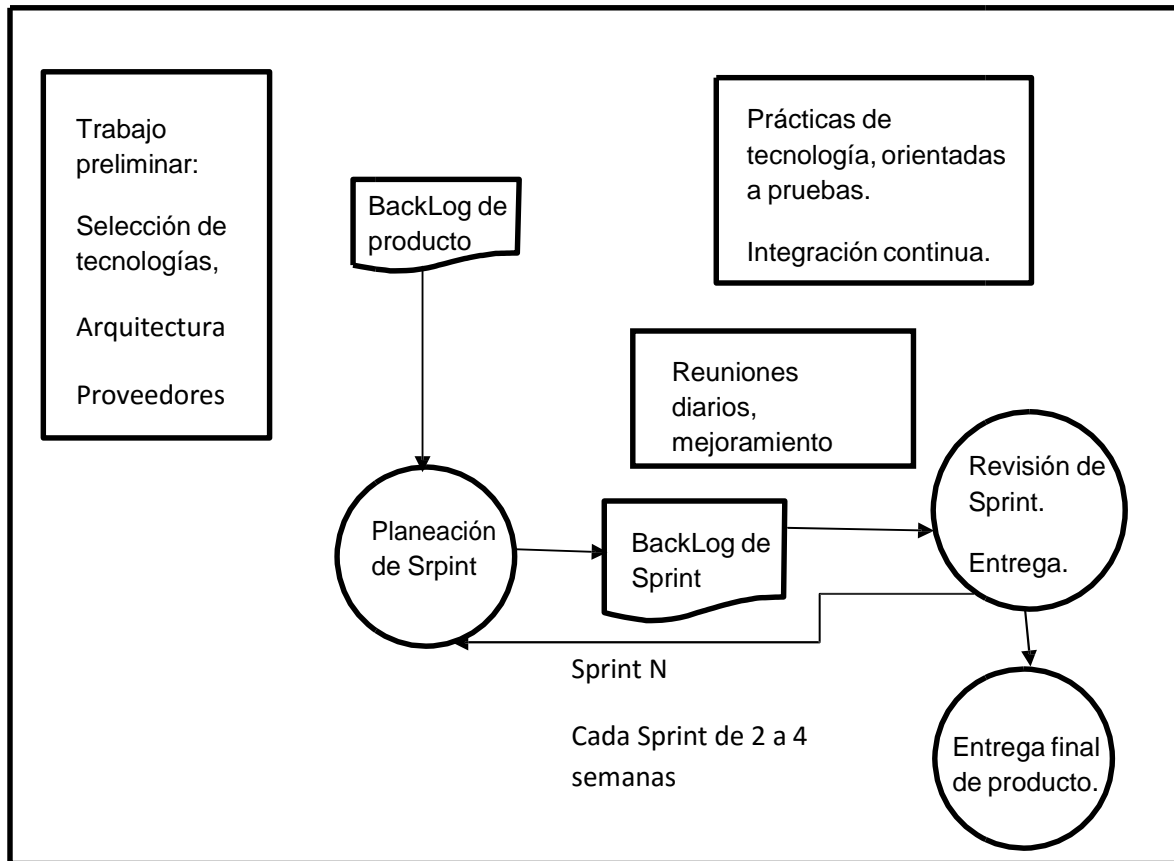


Ilustración 1.4 Scrum adaptado para el proceso de desarrollo de Software ágil. (Yong, 2011)

Uno de los conceptos principales dentro de la metodología descrita en (Mundra, Barclays, & Misra, 2013) es el Back Log, que es un documento dentro del cual se establecen los requerimientos para las funcionalidades esperadas y que posteriormente se dividirá en Sprints donde se define un tiempo de desarrollo para cada iteración o ciclo de desarrollo en la cual se hace una entrega de avances directamente con el cliente y se hacen los registros pertinentes en el Backlog.

Con respecto al número de integrantes para el grupo de desarrollo se propone generar pequeños grupos a los que se les asigna una carga de trabajo de acuerdo con sus capacidades e integrantes.

Tabla 1.2 Estructura de un equipo de SCRUM. (Yong, 2011)

Roles	Comentarios	
	Cantidad	Comentarios
Scrum Master	1	100% solo como Scrum Master o bien 50% como Scrum Master y 50 % como cualquier otro rol.
Propietario del producto	1	Coordina los requerimientos de n números de usuarios finales / clientes y prioriza los requerimientos.
Desarrolladores	Máximo 6	Con más de 6 desarrolladores se debe de considerar dividir el proyecto en más derivantes y hacer Scrum de Scrum.
Tester	Por lo menos 1 por cada 3 desarrolladores.	100% desde el inicio del proyecto

Dentro de este artículo se menciona que los autores eligieron restringir el número de integrantes del equipo desarrollador en 6 debido a que al momento de la reunión del Sprint se tendrá un grupo de 10 personas en la sala de reuniones, conformado por el Scrum master, el propietario del producto, 2 Testers y los 6 desarrolladores. El Scrum Master estará encargado de la cohesión del grupo así como de encargarse de que se cumplan con todos los preceptos de la metodología durante el desarrollo. El propietario de producto es el encargado de escoger las prioridades ya que representa al cliente final presentando así el rol más importante. El Tester es el encargado de ejecutar las pruebas funcionales y deberá de estar involucrado en el proyecto desde el inicio de este.

Scrum de Scrum se refiere al desarrollo de un proyecto dentro del cual se integran múltiples equipos de Scrum orientados a desarrollar el mismo producto. Por ejemplo cuando el proyecto está siendo desarrollado en diferentes ubicaciones y cada una de estas cuenta con un equipo de Scrum.

(Barton, 2009) Uno de los principales objetivos de las empresas hoy en día es presentar un valor agregado a sus productos para captar la atención de los clientes.

La metodología ágil de Scrum permite brindar este agregado en las empresas que lo practican en un corto tiempo. Las pequeñas entregas realizadas en cada sprint aceleran los tiempos de entrega y retroalimentación aumentando la difusión del producto. Scrum facilita el proceso de innovación debido a que para poder implementarse se comienza con un estudio del desarrollo para poder simplificarlo en formas menos complejas que puedan ser correctamente entendidas. Lo cual, en consecuencia lleva a los desarrolladores a innovar aspectos concernientes al nuevo desarrollo. Y aunque, de cierto modo el generar innovación representa riesgos, Scrum ayuda a trabajar esos riesgos al brindar retroalimentación en cada entrega generada por cada iteración.

Tabla 1.3 Alternativas de lenguaje para Scrums que no son desarrolladores de software. Brent Barton (2009)

Lenguaje ágil y de Scrum para Software	Alternativas de lenguaje para Scrums no desarrolladores de Software
Software	<ul style="list-style-type: none"> □ Iniciativa de negocios. □ Proceso de negocios. □ Producto. □ Actividad incrementadora de valores.
Software con potencial para envío	<ul style="list-style-type: none"> □ iniciativa de negocios potencial. □ Proceso de negocios potencialmente desplegable. □ Producto con potencial para lanzamiento. □ Valor de negocios potencialmente implementable.
Entrega	<ul style="list-style-type: none"> □ Entrega □ Despliegue □ Distribución.
Definición de terminado	<ul style="list-style-type: none"> □ Estándar de calidad. □ Checklist de completitud
Aceptación	<ul style="list-style-type: none"> □ Criterios de aceptación
Criterio	<ul style="list-style-type: none"> □ Condiciones e satisfacción. □ Plan para manejo de cambios

Scrum es una metodología tan adaptable que puede ser implementada en cualquier situación que requiera de una planeación seria, en la Tabla 1.3 se puede observar una analogía de conceptos para el mejor entendimiento de estos.

(Khalane & Tanner, 2013) Mencionan algunas de las problemáticas que surgen dentro de la implementación de metodologías ágiles surgen en las carencias de estas mismas, al funcionar como un marco de trabajo generalizado, estas omiten ciertos aspectos que necesitan ser adaptados y completados adaptando la metodología a otras técnicas que puedan eliminar estos vacíos de la manera que mejor se adapte en vías al aseguramiento de la calidad en el desarrollo de software. Cuando se le agrega el concepto de aseguramiento de la calidad al marco de desarrollo de Scrum se aporta un valor agregado a los desarrollos. El aseguramiento de la calidad en el software se enfoca en desarrollar un producto el cual no cuente con fallas y cumpla en los tiempos de entrega brindando de cierto modo un renombre competitivo a la empresa en el sentido de que cumple y excede las expectativas de cada cliente con el que se haya interactuado. El manejo de la calidad se refiere a la búsqueda del aseguramiento de los estándares de calidad al definir un proceso que siga al pie de la letra los lineamientos de calidad para el desarrollo en cuestión arrojando como resultado final; después de la fase de pruebas, productos de software de calidad.

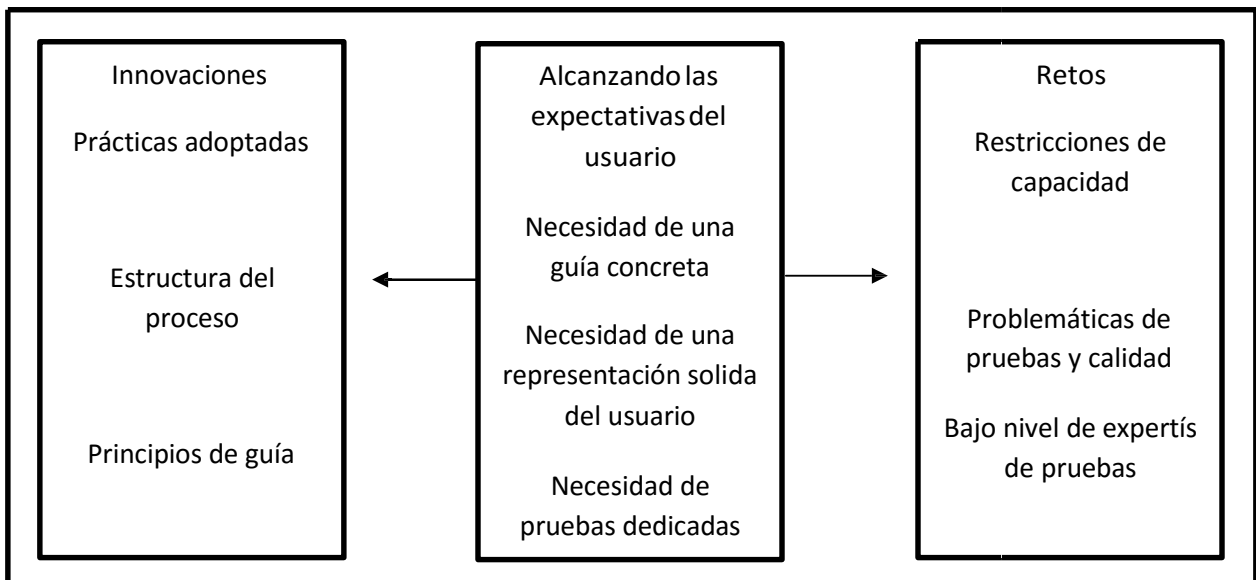


Ilustración 1.5 Aspectos de aseguramiento de la calidad en un ambiente de desarrollo con Scrum (Khalane & Tanner, 2013)

Como puede observarse en la Ilustración 1.5 la metodología Scrum se queda corta cuando se enfoca hacia el aseguramiento de la calidad del software, pero a su favor muestra una gran facilidad para la adaptabilidad y cohesión con otras técnicas que la complementan en donde encuentra fallas. La mejor solución a esta problemática es crear una guía de aspectos técnicos para alcanzar el aseguramiento de la calidad. Esto se logra motivando a los desarrolladores a buscar innovación incluyendo y

adaptando prácticas de otras metodologías incorporándolas a Scrum de tal manera que se alcancen los estándares de calidad brindando una mejora continua a los procesos de desarrollo dentro de la misma empresa.

(Sutherland & Patientkeeper, 2005) La metodología ágil Scrum ha tenido un gran impacto en los desarrollos contemporáneos gracias a su adaptabilidad, a la manera en que da rapidez a los desarrollos y los resultados que brindan total satisfacción al cliente al interactuar con el dentro de cada iteración del proceso. A pesar de todos estos resultados Scrum sigue evolucionando siempre en busca de mejoras, para poder brindar estos cambios se trabaja bajo una ideología de analizar lo que te ha funcionado y podría ser útil para implementar en el futuro, además de identificar que entorpece el desarrollo para poder eliminarlo del proceso (teoría de Scrum, evolución de Scrum, preconcepciones de Scrum; respectivamente). Al agilizar los procesos en la implementación en la metodología Scrum, la empresa se vuelve capaz de hacer entregas de producto funcional aún más rápido que la posible demanda de este; eliminando todas las posibilidades de retrasos y problemáticas dando oportunidad a que la empresa pueda desarrollar nuevos y más complejos productos con más valores agregados que antes no le era permitido.

Scrum puede ser dividido en tres tipos de acuerdo a sus iteraciones:

El primer tipo "Tipo A" es el que se encuentra marcado por iteraciones aisladas:

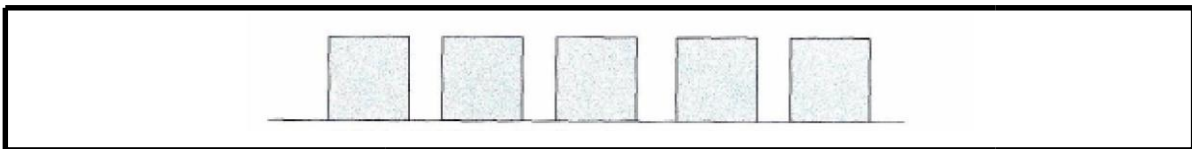


Ilustración 1.6 Iteraciones aisladas. Jeff Sutherland (2005)

Este tipo de Scrum se utiliza generalmente dentro de empresas que comienzan con la implementación de esta metodología o con propósitos académicos.

El "Tipo B" es aquel en que las iteraciones se encuentran traslapadas.

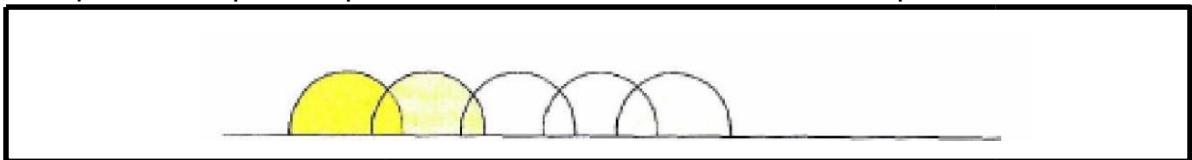


Ilustración 1.7 Iteraciones traslapadas. Jeff Sutherland (2005)

En esta implementación se atacan las fallas encontradas en el tipo A al generar tareas que pueden ser integradas dentro de las primeras fases de implementación del siguiente sprint.

El Scrum de "Tipo C" se enfoca en realizar las iteraciones todas a la vez.

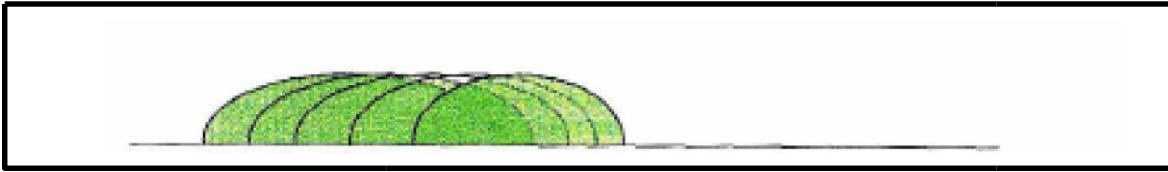


Ilustración 1.8 Todas a la vez. Jeff Sutherland (2005)

Este tipo de implementación es implementada mayormente en desarrollos en los cuales una actividad es altamente impredecible, el enfoque cambia de dirección en repetidas ocasiones y se agregan tareas a medida que el proyecto se va desarrollando.

(Fan, Cheded, & Toker, 2005) Java es uno de los lenguajes de programación con más aplicaciones para internet hoy en día, debido a que este lenguaje proporciona portabilidad de código: es decir, las aplicaciones desarrolladas en Java permiten su ejecución bajo una máquina virtual sin importar el sistema operativo. XML es una especificación propuesta por la W3C (World Wide Web Consortium) para realizar una descripción estructurada de información, proporcionando métodos efectivos y eficientes para el almacenamiento y compartición de la información; es decir, portabilidad de los datos. Estas dos tecnologías pueden interactuar de diversos modos; siendo el más básico el de acceder a información de los archivos XML desde la programación de Java; pero esto involucra tener conocimientos acerca de cómo poder hacerlo. El primer enfoque para poder hacerlo es SAX (Simple API for XML) que funciona como un flujo de datos que obtiene los datos directamente del XML fuente. El segundo es DOM (Document Object Model) el cual permite realizar un modelado de los datos manteniendo el documento en la memoria.

JAXP (Java API for XML Parsing) presenta una tercera alternativa mediante la cual se abstraen los datos obtenidos del documento fuente.



TECNOLÓGICO NACIONAL DE MÉXICO
Instituto Tecnológico de Apizaco

"2015. Año del Generalísimo José-María Morelos y Pavón"



CAPITULO II FUNDAMENTOS

REDES Y DISPOSITIVOS INTELIGENTES

(Momoh, 2012) Menciona en su trabajo que una red inteligente o Smart Grid es un sistema integrado por dispositivos digitales encargados de conservar la confiabilidad y eficiencia de este de manera automatizada con un flujo de comunicación de dos vías, se auto repara, es adaptativo y es sustentable; cuenta además con un enfoque de predicción bajo incertidumbre.

Es equipado para interoperabilidad con estándares para componentes, dispositivos y sistemas cibernéticamente seguros ante ataques maliciosos ya sean actuales o futuros, estos dispositivos de medición realizan sus tareas en tiempo real.

Tabla 2.1 Comparación de la red actual contra la red inteligente. (Momoh, 2012)

Características preferidas	Red actual	Red inteligente
Participación activa del usuario	Regularmente usuarios no informados y participan.	los no son no Informativa, involucre a los usuarios, responde a la demanda y tiene Fuentes de energía distribuidas.
Alojamiento de toda la generación y opciones de almacenamiento	Dominado por la generación existen obstáculos para la interconexión para fuentes de energía distribuidas.	por la central, algunos para la para Fuentes de energía distribuidas con un enfoque de Plug & Play y hacia la renovación.
Nuevos productos, servicios y mercados	Limitado, mercados pobremente integrados, oportunidades limitadas para los consumidores.	Madura, mercados bien integrados, además del crecimiento de mercados de electricidad para los consumidores.
Provisión de calidad de energía para la economía digital	Se enfoca en los apagones, con respuesta lenta hacia asuntos de calidad de energía.	La calidad de energía es una prioridad con una variedad de opciones en calidad / precio, resolución rápida de acontecimientos.

Optimización de bienes y eficiencia de operativos	Poca integración de datos operacionales con el manejo de bienes.	Adquisición de datos de parámetros de la red ampliamente expandida, se enfoca en la prevención, minimizando el impacto en los consumidores
Anticipación de respuestas al Sistema de problemáticas (Self-healing)	Responde para prevenir daño futuro; enfocándose en protección de bienes tras una falla.	Detecta y responde de manera automática a los problemas, se enfoca en la prevención minimizando el impacto en los consumidores
Resistencia en contra de cyber ataques y desastres naturales	Vulnerable hacia actos maliciosos y desastres naturales, respuesta lenta.	Resistente a cyber ataques y desastres naturales; capacidades de restauración rápida.

Algunos de los conceptos involucrados en el desarrollo de una inteligente según (Momoh, 2012). Son:

- Inteligencia computacional.
- Implementación de herramientas analíticas usadas para la optimización de los dispositivos que la integran, haciendo uso de técnicas como heurística, programación evolutiva, etc.
- Mejoramiento del sistema eléctrico.
- Incluir tecnologías generadoras de energía renovable.
- Estándares de comunicación.

La comunicación de los dispositivos que integran la red de ser capaces de tomar decisiones de manera anticipada debido a los datos generados en tiempo real que viajan a través de esta.

(Kenneth C. Budka, 2014) Describe al Head End como el sistema manejador de las infraestructuras avanzadas de medición o AMI (Advanced Metering Infrastructure). Se comunica con el medidor concentrador por medio de una conexión IP provista por la red inteligente bajo estándares de comunicación establecidos por el proveedor. El Head End también mantiene comunicación con la Utility con una conexión IP bajo una LAN.

ACCESO A ARCHIVOS XML.

Con respecto al intercambio de información con los medidores se hace uso de documentos con extensión XML. Cabe resaltar que el sistema de la Utility no se comunica directamente con los medidores, para poder enviar información hacia un medidor específico este paquete debe de ser enviado hacia el Head End.

(Syngress, 2002) Para todo método interprete de archivos XML debe de haber una entrada, esta se refiere al documento a ser accedido, el cual puede ser ingresado como una cadena o una ruta de almacenamiento. Para la interpretación de archivos XML con DOM (Data Object Mapping) la primer tarea a realizar es la obtención de la raíz del documento o bien nodo raíz debido a que una vez que se encuentra este nodo el procesamiento del árbol DOM puede ser iniciado. El siguiente método descrito se encargará de analizar la jerarquía del árbol explorando el árbol de manera que se accesa al nodo y a todas sus hojas hasta alcanzar el final para poder continuar con el siguiente nodo si es que existe, repitiendo esta operación tantas veces sea necesario considerando los niveles de estructura del archivo procesado.

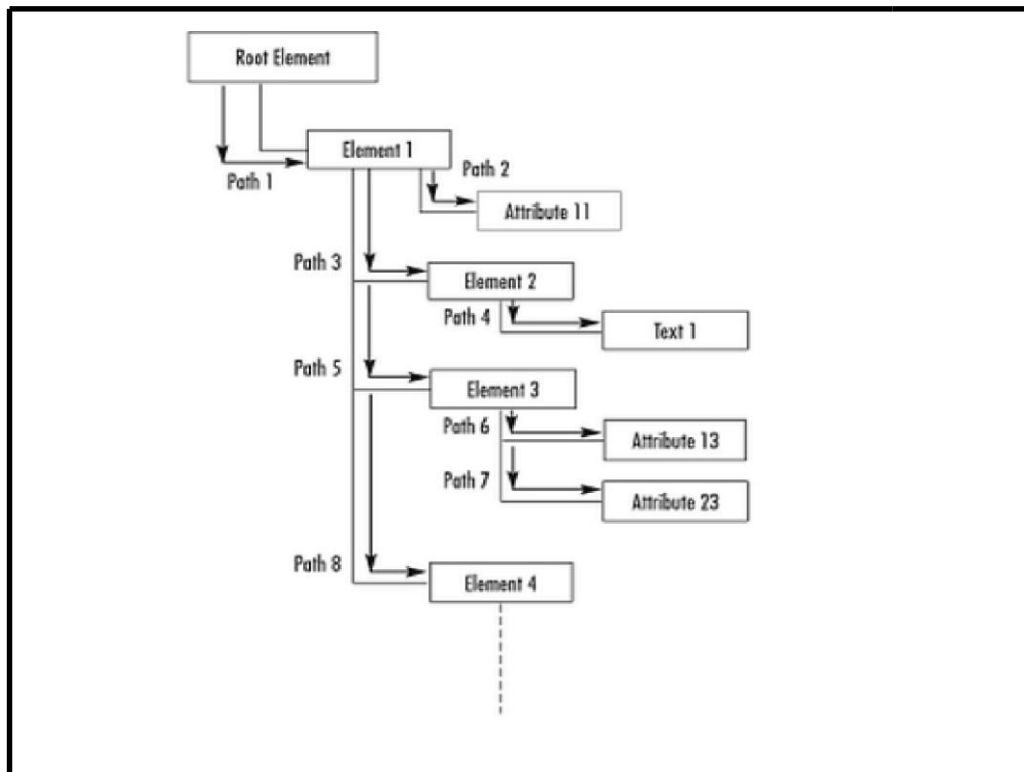


Ilustración 2.1 Accesando al árbol DOM.

En la ilustración 2.1 se pueden observar los accesos a los nodos considerando las jerarquías dentro del árbol DOM de cualquier método lo cual brinda un algoritmo para el desarrollo de una función encargada de realizar la lectura de estos documentos.

METODOLOGÍAS Y TÉCNICAS DE PROGRAMACIÓN

El manejo de un proyecto es un concepto que se debe de tener en cuenta en todo momento al inicio de cualquier desarrollo; ya que a partir de este se proyectará la gestión de recursos y distribución de la carga de trabajo para los integrantes del equipo de desarrollo.

(Institute, 2009) "Un proyecto es un esfuerzo temporal que se lleva a cabo para crear un producto, servicio o resultado único".

Los proyectos generados dentro de las organizaciones tienen siempre como finalidad el cumplir metas propuestas en cuanto a un desarrollo, la situación óptima esperada en la realización de un proyecto es que este llegue a una conclusión satisfactoria, es decir la generación de un producto, servicio o cualquier otro tipo de resultado planteado el cual puede cumplir con la satisfacción de una necesidad dentro del mercado, de un cliente específico, una actualización o mejora; pero para poder lograrlo la planeación estratégica del desarrollo tiene un impacto primordial.

Dentro de la planeación estratégica para el proyecto se deberán considerar primeramente los requisitos específicos del desarrollo tomando en cuenta restricciones tales como tiempo, costos, recursos, etc.

Con la finalidad de realizar una gestión de recursos de manera exitosa. En algunos casos las limitaciones para el desarrollo del proyecto comienzan definidas por normatividades bajo las cuales está regido el producto o servicio, por lo cual estas normas o estándares se deben de tener bien presentes desde la fase de planeación, estas regulaciones garantizan que el resultado final cumplirá con todos los lineamientos de calidad.

Existe una serie de conceptos que se deben de tener en cuenta para una óptima dirección de proyectos, los cuales en conjunto con todos los preceptos anteriormente mencionados llevarán a una conclusión satisfactoria del desarrollo:

Tabla 2.2 Habilidades interpersonales necesarias para la gestión de proyectos. (Institute, 2009)

habilidades interpersonales
Comunicación efectiva.
Influencia en la organización
Liderazgo
Motivación
Negociación y gestión de conflictos
Resolución de problemas

Las habilidades mencionadas en la tabla 2.2 son básicas para el personal involucrado en la dirección del proyecto. La correcta interacción de los directores con el personal de desarrollo es la base de un buen desarrollo, para poder relacionarse de manera correcta con el resto del equipo los encargados de la gestión del proyecto deben de comunicar los objetivos de manera clara y oportuna expresándolos de una manera entendible para todo el personal asignando y delimitando de manera eficiente las tareas a tiempo para que estas puedan ser realizadas, el liderazgo es primordial; un buen líder muestra a sus colaboradores la manera de realizar las tareas de manera correcta tomando la iniciativa manteniendo la motivación constante durante el ciclo de vida del proyecto.

Los conflictos son una situación natural en las interacciones diarias de una empresa por lo cual el personal de dirección debe de contar con las habilidades interpersonales para poder solucionar los posibles problemas que puedan surgir evitando la generación de conflictos que puedan surgir y retrasar o complicar de cualquier modo el desarrollo del proyecto.

(Rubin, 2013) Scrum no es un proceso estandarizado donde pueda usted metodológicamente seguir una serie de pasos que garanticen la producción de productos a tiempo y dentro del presupuesto que deleiten al cliente. En lugar de esto, Scrum es un Framework para organizar y manejar el trabajo. El Framework de Scrum está basado en un conjunto de valores, principios, y prácticas que proveen los fundamentos para que su organización pueda agregar le implementación de prácticas de ingeniería relevantes y enfoques específicos para completar las prácticas de Scrum; el resultado será una versión de Scrum única.

Esta metodología está basada en un artículo llamado "El juego de desarrollo del nuevo nuevo producto" publicado en 1986 en el cual se describe como compañías de gran renombre implementaron un enfoque escalable orientado a un desarrollo de equipos para generar productos de clase mundial; aunque fue creada hasta 1993 por

Jeff Stherland y su equipo para uso en el desarrollo de software, y aunque esta metodología es más usada dentro del campo de desarrollo de software, sus principios más básicos pueden ser usados para desarrollar una gran variedad de productos.

Entre los beneficios que aporta esta metodología se puede encontrar:

- Satisfacción al cliente.
- Mejora en la devolución de la inversión.
- Reducción de costos.
- Producción rápida de resultados.
- Brindar confianza a la empresa para competir a nivel mundial.
- Brindar comodidad a los empleados.

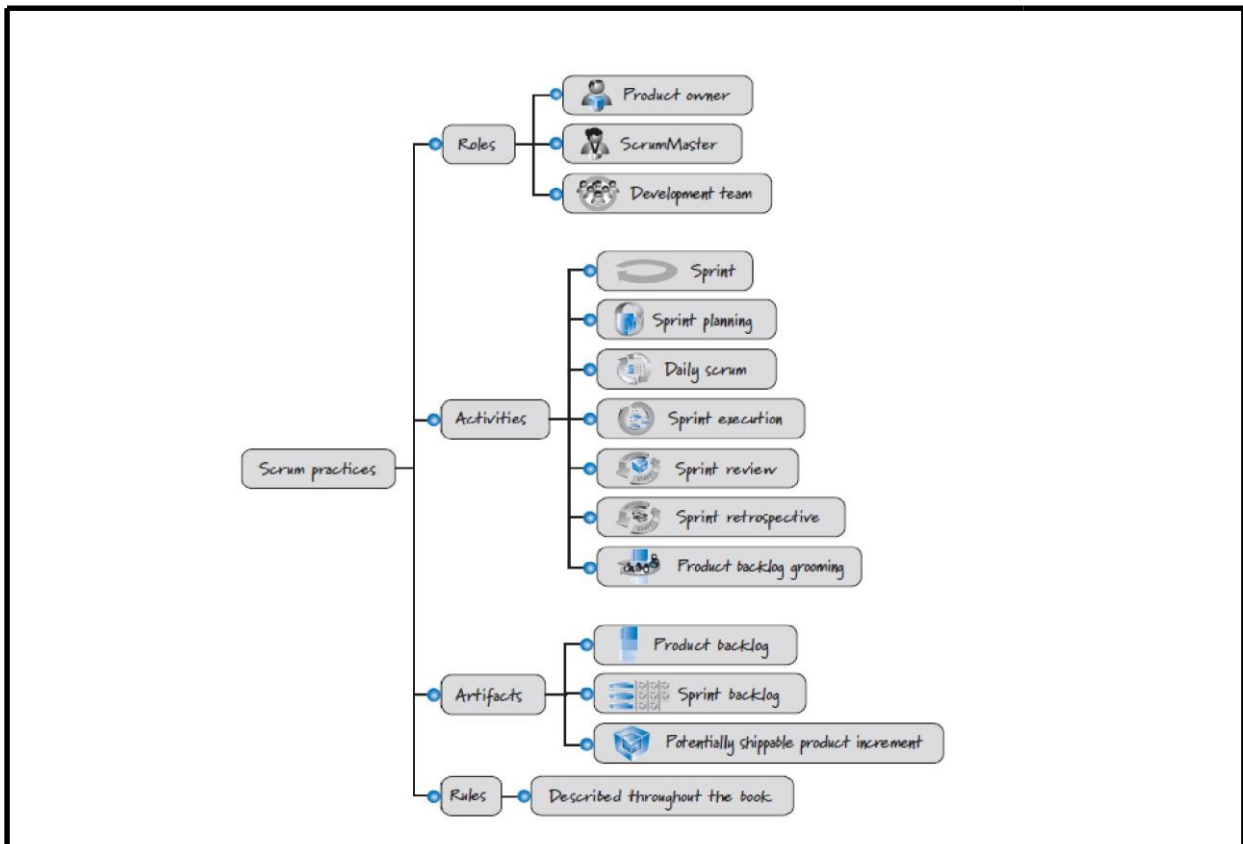


Ilustración 2.2 Prácticas de SCRUM. (Rubin, 2013).

En la ilustración 2.2 se pueden observar las prácticas para implementar de manera general, observando una división de roles, actividades, artefactos y reglas.

Los roles dados dentro de la metodología tienen como objetivo la integración de equipos de trabajo que cuenten con los tres principales roles enlistados de manera básica; los cuales son, el propietario del producto, el Scrum Master y el equipo de desarrollo.

El propietario del producto es el encargado del liderazgo dentro del grupo, es la principal autoridad y el responsable de las decisiones que impactan directamente dentro el desarrollo del proyecto.

El Scrum Master actúa a manera de manejador o entrenador brindando a los integrantes del proyecto un enfoque de trabajo orientado a Scrum inculcándoles las prácticas, principios y valores en los que se basa la metodología, de igual manera puede aportar ayuda en la resolución de problemas dentro del grupo de trabajo así como proteger de posibles interferencias o impedimentos externos a este.

El equipo de trabajo, en el estarán integrados todos los involucrados en el desarrollo del producto final, a cada integrante de este se le deberán de asignar las actividades organizándolas de la mejor manera para poder completar los objetivos impuestos por el propietario del producto. Los grupos de trabajo dentro de la metodología están considerados para ser de 5 a 9 integrantes.

Un sprint dentro de Scrum es un ciclo calendarizado, en el cual deberá de entregarse como producto final un avance tangible del desarrollo al cliente. La planeación de un Sprint acompaña el desarrollo del mismo y termina en una revisión y retrospectiva. La planeación funciona como una predicción o un compromiso donde se proponen las metas a alcanzar en cada entrega. De manera diaria se habrá de realizar un Sprint diario, integrado por los miembros del equipo de desarrollo, donde se reúnen para presentar asuntos referentes al proyecto como avances, problemáticas, impedimentos, y planeaciones. En cuanto a la planeación del desarrollo en general, se consideran conjuntos de características que el producto final habrá de contener una vez terminado basados en un análisis de requerimientos, a los cuales les es asignada una prioridad en relación a su importancia según la considere el jefe del equipo. De acuerdo a la priorización de las tareas se asignan cargas de trabajo a cada integrante del equipo y fechas de entrega correspondientes a los Sprints establecidos; con la finalidad de cerciorarse del avance obtenido así como la posible actualización de alguna de las características o modificación de un requerimiento. Esto último permite un diseño incremental adaptativo en el cual se asegura la satisfacción del cliente y la calidad del producto final. Al final de cada Sprint se realizan tareas de revisión en las cuales se busca realizar una buena adaptación del producto, lo cual resultará en un equipo de desarrollo consiente de las incidencias y

mejoras que se puedan implementar ya que ha recibido retroalimentación directamente del cliente o propietario del producto, lo cual es casi una garantía de que se alcanzaran a cubrir las expectativas de este último en la entrega del producto final.

La programación Extrema fue implementada del mismo modo dentro del desarrollo del proyecto, el conocer los conceptos básicos y estar familiarizado con la técnica brinda madurez y agilidad a los procesos de programación.

(Chromatic, 2013) Este enfoque de desarrollo encuentra sus fundamentos en término de 4 variables, tiempo, alcance, recursos y calidad. Cada proyecto trabaja bajo diferentes combinaciones de estas variables, incluso si estas no son medidas de manera directa, sus interacciones forman el proyecto.

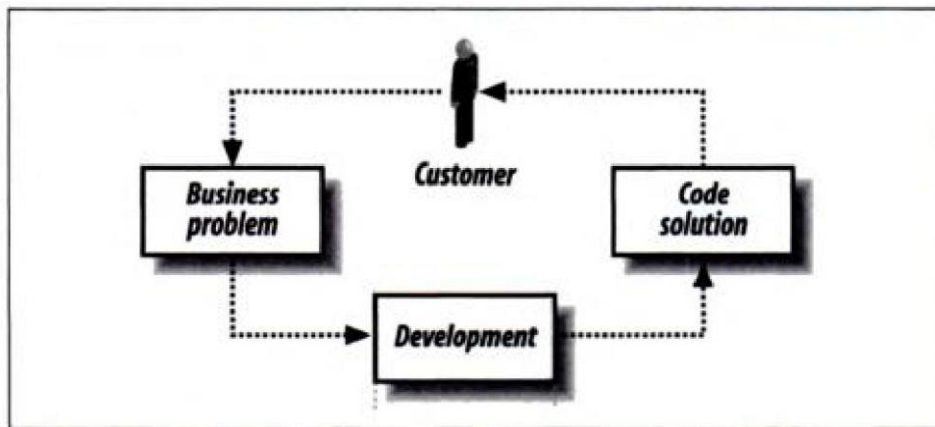


Ilustración 2.3 Ciclo de la programación extrema.(Chromatic, 2013)

Para el ciclo dentro de la programación extrema podemos observar en la Ilustración como comienza con las requisiciones del cliente representado como el problema de negocio, la cual es desarrollada e implementada por los desarrolladores que forman parte del equipo. Una vez terminada, la solución de código es entregada al cliente; basado en los resultados percibidos por el cliente, este podrá hacer otras peticiones o aclaraciones y de esta manera el ciclo continuará iterando hasta llegar a la solución principal.

(Kent Beck, 2005) La programación Extrema es un estilo de desarrollo de software enfocado en técnicas de programación, comunicación limpia y trabajo en equipo lo cual permite alcanzar metas inimaginables.

Extreme Programming es:

- una filosofía de desarrollo de software basada en los valores de comunicación, retroalimentación, simplicidad, coraje, y respeto.
- Un cuerpo de prácticas que ha sido probada útil para el mejoramiento del desarrollo de software. Las prácticas se complementan una a la otra, amplificando sus efectos.
- Una comunidad que comparte valores y muchas de las mismas prácticas.

La programación extrema puede ser descrita como:

- XP es ligera, en la programación extrema solo se hace lo que es necesario para crear algo de valor para el cliente.
- Es una metodología basada en referirse a las limitantes en desarrollo de software.
- Puede ser útil en equipos de cualquier tamaño.
- XP se adapta a requerimientos vagos o que cambian constantemente.

(Beck & Fowler, 2000) La planeación es uno de los aspectos principales dentro del desarrollo software, ya que brinda un panorama de las eventualidades que se pueden presentar, además de controlar los recursos a disposición del proyecto, así como priorizar las actividades.

Al comienzo de cualquier desarrollo la primer actividad que se debe desarrollar es la generación de actividades de acuerdo a los requerimientos establecidos para el proyecto; para cumplir con cada necesidad del cliente se deberá de generar una actividad la cual implica el uso de recursos propios de la empresa por lo que es de gran importancia realizar una planeación lo más cercana a la realidad (habilidad que solo es obtenida por la experiencia) para que; una vez que hayan sido generadas las tareas a desarrollar, estas sean priorizadas para asegurar que no se perderá tiempo en actividades que pueden ser reemplazadas o eliminadas mientras el proyecto avanza.

Para lograr una correcta disposición de recursos para el desarrollo, se debe de tener completa conciencia de sus características o habilidades; el jefe de proyecto debe de conocer las capacidades de cada uno de sus empleados para poder asignarle tareas acorde a su nivel de desempeño; del mismo modo un ordenador debe de cumplir con los requerimientos mínimos para poder desempeñar las actividades asignadas en él.

Tabla 2.3 Comparativa de miedos y derechos de clientes y desarrolladores.

Miedos		Derechos	
Cliente	Desarrollador	Cliente	Programador
No obtener lo solicitado.	Que se les asignen tareas de las que no tienen conocimientos.	A recibir un plan general y a saber que será terminado, cuando y a que costo.	A saber qué es lo que se necesita, y sus prioridades.
Solicitar lo equivocado.	Que se les asignen tareas que carecen de sentido.	A recibir el máximo de avances en una semana de programación	A producir trabajo de calidad cada vez.
Pagar demasiado por algo pequeño.	Que no tengan el nivel necesario.	Ver el progreso en un sistema funcional el cual deberá de pasar por pruebas propuestas por el cliente mismo.	A solicitar y recibir ayuda de compañeros, gerentes y clientes.
Ceder ante personas sin el mismo interés.	Que se estén volviendo obsoletos.	A cambiar de opinión, sustituir funcionalidad y cambiar prioridades sin pagar sumas exorbitantes.	A realizar y actualización sus propios estimados.
Jamás ver un plan significativo.	Que se les asignen responsabilidades sin darles autoridad.	A estar informado de cambios en las planeaciones a tiempo para poder escoger como reducir el tiempo para poder alcanzar las fechas establecidas. Pueden cancelar en cualquier momento y obtener aun así un sistema funcional que	A aceptar responsabilidades en vez de que le sean asignadas.

refleje la inversión a la fecha.	
Los planes que le son presentados son irrealistas.	Que no se les den instrucciones claras de lo que deben desarrollar.
No saber lo que está sucediendo.	Que tengan que sacrificar calidad por proximidad de fechas límites.
Quedar anclados a decisiones previas sin opción a adaptación a cambios en el negocio.	Que tengan que solucionar problemas difíciles sin ayuda.
Que nadie les diga la verdad.	Que no cuenten con tiempo suficiente para obtener éxito.

Durante el proceso de planeación de todo proyecto se debe de contemplar los miedos tanto del cliente como de los desarrolladores; esto con la finalidad de ofrecer una completa seguridad y transparencia hacia el cliente de lo que se trata el proyecto, lo que se acuerda realizar y los tiempos para completarlo, así como de informar al equipo desarrollador de cuáles son los requerimientos específicos solicitados y el alcance de su participación y tiempo estimado de desarrollo para que estos estén conscientes de su carga de trabajo y responsabilidades.

(Layka, 2014) Existe una gran diferencia entre una aplicación web y una página Web, una Webapp está diseñada para generar información a partir de la interacción con el usuario mientras que un sitio web es más estático y generalmente solo sirve para la presentación de información al usuario. Java cuenta con la plataforma EE para el diseño de aplicaciones Web la cual brinda una serie de herramientas enfocadas a facilitar el desarrollo de este tipo de sistemas al desarrollador.

En la ilustración 2.4 se puede observar la arquitectura multicapa en java la cual comienza con la capa del cliente, que se refiere a la aplicación mediante la cual el usuario interactúa; es decir, muestra una interfaz mediante la cual el usuario puede

generar las peticiones que procesará el servidor en una capa diferente de esta arquitectura.

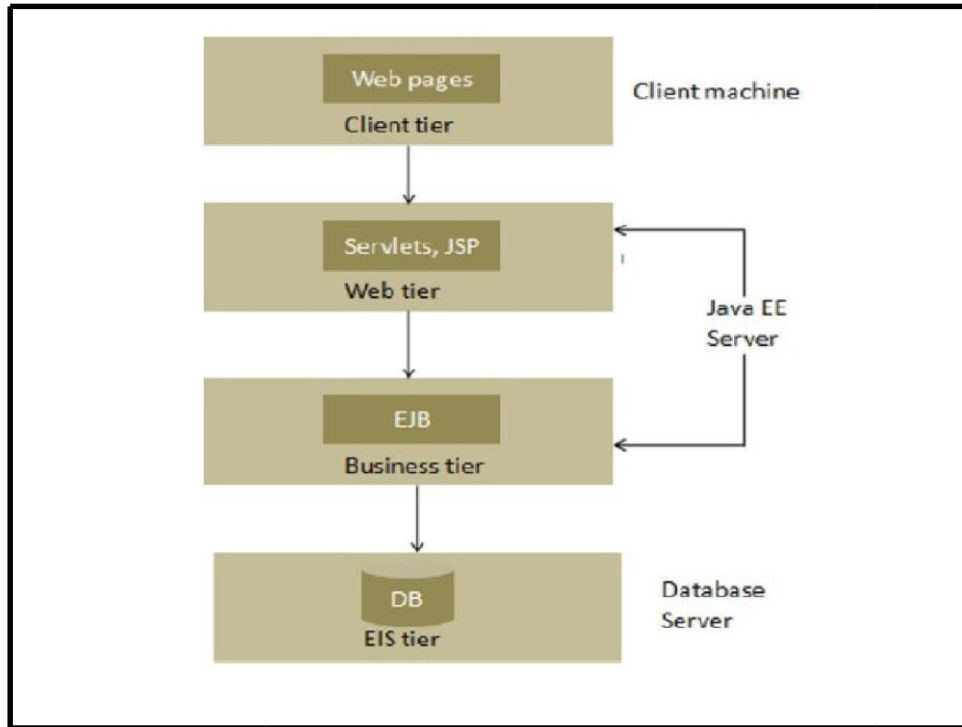


Ilustración 2.4 Arquitectura multicapa en Java.(Layka, 2014)

La capa Web se refiere a la funcionalidad encargada de juntar la funcionalidad de la capa del cliente con la de la capa de negocios. Esta brinda la interacción mediante la cual el cliente puede ingresar información, mantener los datos actualizados del usuario dentro de la sesión y generar contenido de acuerdo a los datos ingresados por el usuario.

(Englander, 1997) Los componentes son elementos auto contenidos de software que pueden ser controlados dinámicamente y ensamblados para formar aplicaciones. Los componentes además pueden inter operar de acuerdo a un conjunto de reglas y lineamientos. Los Java Beans son componentes de modelo. Permiten al usuario construir aplicaciones ensamblando componentes ya sea de manera programada, visual o ambas. La arquitectura de los JavaBeans soporta características de software como reutilización, modelos de componentes, y son orientados a objetos; además de permitir que los valores de sus propiedades sean cambiados desde una interfaz visual.

Debido que los JavaBeans con desarrollados en java son completamente portables a cualquier plataforma donde se ejecute el Java Run-time Environment incluyendo exploradores web, dispositivos móviles, etc.

(Brown, Dalto, & Jepp., 2011) El lenguaje de programación java en su edición empresarial (Enterprise Edition) cuenta con dos tecnologías referentes a la producción de contenido Web. Los Servlets incrustan contenido dentro de la programación haciendo uso de clases de Java con lo cual procesan las peticiones generadas por el cliente usando uso de HTTP y generan de manera dinámica una respuesta adecuada en forma de HTML o XML; mientras que JSP (Java Server Pages) hace exactamente lo opuesto presentándose en forma de HTML, WML, XML, la petición generada en este es manejada en primera instancia por el servidor web; el que después la envía hacia un Servlet de implementación.

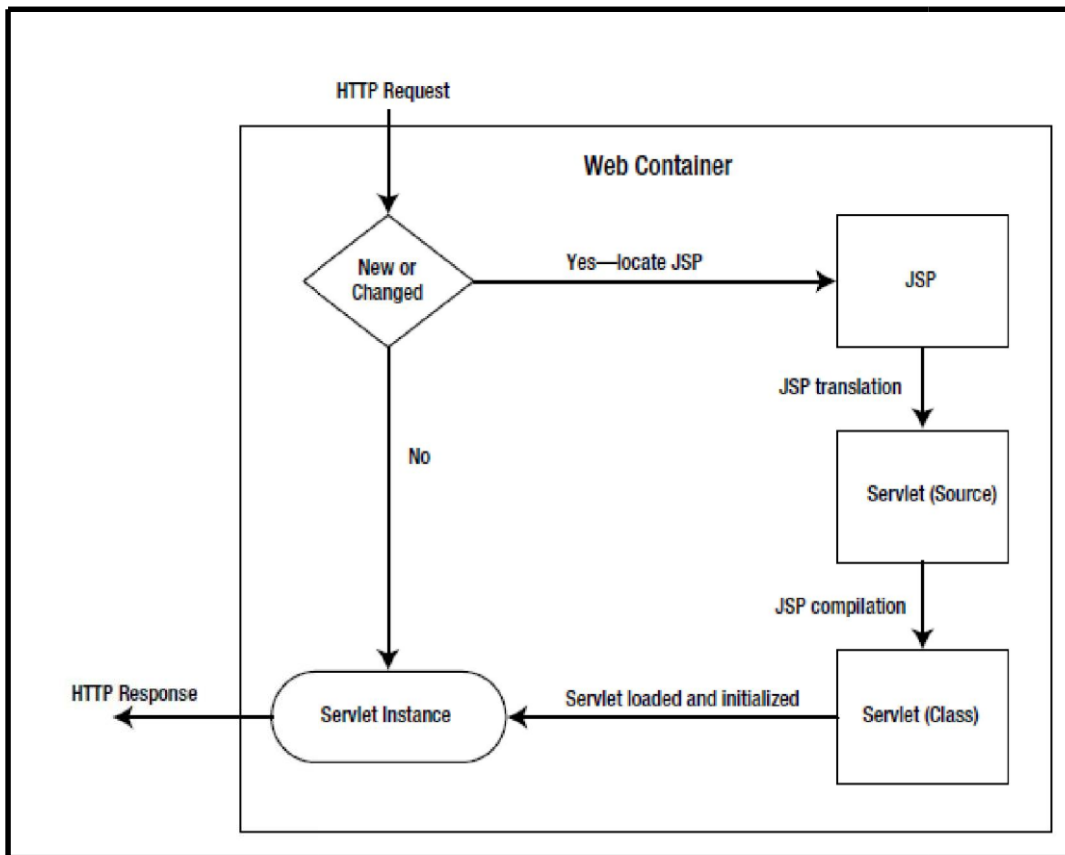


Ilustración 2.5 Flujo de peticiones dentro de un JSP.(Brown, Dalto, & Jepp., 2011)

“Antes de procesar la petición, el contenedor determina si es que la fuente del JSP es nueva o ha cambiado; en caso afirmativo, el contenedor traduce y compila la página JSP en una clase Servlet, o una clase de implementación de página, antes de

pasar la petición hacia el Servlet para procesarla". Dentro del ciclo de vida de un JSP encontramos de inicio la fase de traducción, en esta fase el motor del JSP se encargará de realizar un análisis sintáctico en busca de cualquier error; en caso de que la petición cumpla con todas las especificaciones, esta será transformada a un Servlet el cual será compilado nuevamente por el motor para poder ser montado en su contenedor para la fase de inicialización. Para poder ser ejecutado se ejecuta el método `_jspService()`, el cual se encarga de manejar la petición para poder generar una respuesta acorde a la solicitud realizada por el usuario; para que finalmente se ejecute el método adecuado para la destrucción del JSP.

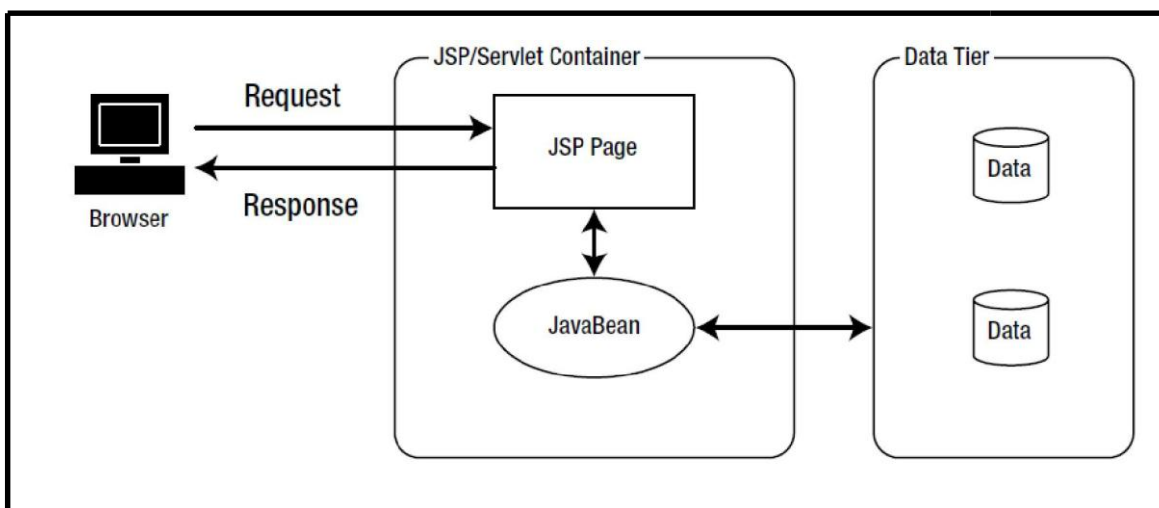


Ilustración 2.6 Arquitectura de un JSP.(Brown, Dalto, & Jepp., 2011)

En la ilustración 2.6 podemos observar el modelo 1 de arquitectura para JSP el cual se caracteriza por que dentro de este el procesamiento ocurre en una sola capa, facilitando su implementación.

JAVA SERVER FACES Y PRIME FACES

(Reddy, 2013) Menciona que Java server faces es un Framework orientado a componentes para desarrollar aplicaciones web usando Java. Facilita la construcción de estas al automatizar tareas comunes como generando parámetros para valores de entrada en Java Beans al transformar parámetros de entrada, ejecutar validaciones y renderizar o actualizar las vistas. Pero este Framework encuentra sus debilidades al momento de la programación integrando JavaScript y HTML para desarrollar interfaces agradables visualmente. Aquí es donde se integra Prime Faces como una alternativa de solución para brindar rapidez y facilidad de implementación al desarrollador; PrimeFaces es soportado en los exploradores más usados (Opera,

Safari, Firefox, Chrome, etc.), cuenta con cientos de componentes construidos para interfaces de usuario que no requieren ser configurados ni necesitan de librerías de terceros para poder ser ejecutados. En un entorno de desarrollo el Framework provee al desarrollador de elementos comunes de entrada de datos extendidos de JSF pero brindando características adicionales con temas visuales mejorados y más atractivos.

(Hlvats, 2013) Describe al framework Prime Faces como una tecnología para desarrollar aplicaciones Web con JSF (Java Server Faces). proporciona un amplio rango de componentes para interfaces de usuario tales como paneles, graficas, botones, links, tablas, Grids, calendarios, cuadros de dialogo, entre otros. Prime Faces es una librería que puede ser integrada a cualquier aplicación de Java Server Faces con una configuración mínima lo cual brinda rapidez a cualquier desarrollo. Los componentes pueden ser agregados en básicamente cualquier parte de la aplicación adaptándolos acorde a las necesidades y requerimientos gracias a que está construido bajo código Open Source.



Ilustración 2.7 Algunos de los componentes ofrecidos por PrimeFaces. (Hlvats, 2013)

PrimeFaces ofrece una gran cantidad de controles listos para usarse con funcionalidades extendidas en comparación con los controles normales de JSF; además de que estos controles pueden ser implementados en desarrollo de aplicaciones para móviles.

SERVICIOS DE WINDOWS

(Robinson, 2004) Un servicio de Windows o Windows Service es una aplicación diseñada para ser ejecutada de manera local en el ordenador en segundo plano.

Por lo general son usados para desarrollar aplicaciones que son ejecutadas de manera continua en espera de algún evento que desencadene su funcionalidad sin la necesidad de ser iniciadas por el usuario; es decir, pueden ser inicializadas una vez que el usuario inicie sesión en el ordenador o inclusive pueden ejecutarse desde que el sistema operativo es Bootado sin necesidad de un inicio de sesión. Para poder observar el estado de la ejecución de un servicio puede ejecutar la herramienta de administración de servicios propia de Windows que forma parte de las herramientas administrativas del sistema.

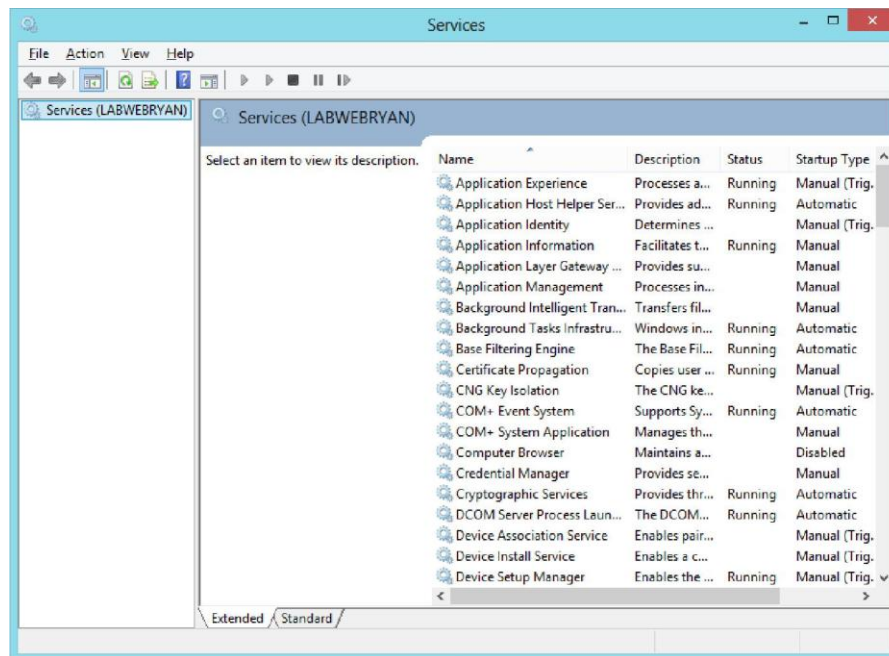


Ilustración 2.8 Administrador de Servicios de Windows.

Nota: Ryan Grisham (November 19, 2012) "Services MMC on Windows 8" Recuperado de: <http://blog.randait.com/2012/11/services-mmc-on-windows-8/>.

En el administrador podemos observar el nombre del servicio una breve descripción, el estatus, además del tipo de inicio que tiene configurado (ilustración 2.8).

Un servicio de Windows cuenta con tres programas para poder operarlo, el programa o servicio que es la aplicación que se ejecuta en segundo plano, el programa de control encargado de iniciar, detener y pausar el servicio en cualquier momento enviándole una solicitud de operación al servicio, además de un programa de configuración o instalador de la aplicación que le indica su tipo de inicio, estado de ejecución de inicio, etc.

JSP Y SERVLETS

(Patzer, 2002) Las aplicaciones Web de java están estructuradas bajo modelos bien establecidos con la finalidad de que estas sean comprensibles; uno de estos modelos es el de Servlet, el cual contenía todo el código de navegación dentro de la aplicación, al igual que la lógica, y la interfaz.

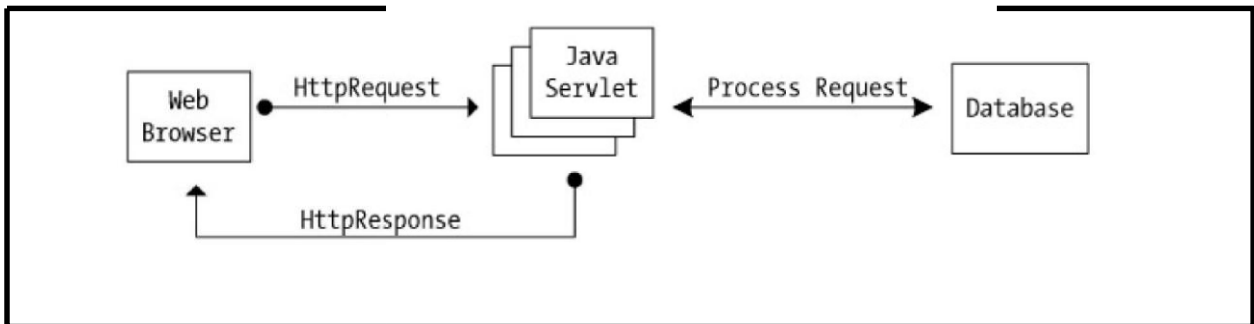


Ilustración 2.9 Modelo de Servlet.(Patzer, 2002)

Como puede observarse en la ilustración 2.9 en el modelo de Servlet el usuario ejecuta una petición desde su explorador web, lo cual inicia una petición que es enviada haciendo uso de HTTP para comunicarse directamente con el Servlet el cual está encargado de ejecutar toda la lógica y comunicarse en caso de ser necesario con la base de datos para poder generar una respuesta coherente a la solicitud del usuario.

En la actualidad la mayoría de las aplicaciones aun funcionan bajo el modelo 1 de JSP el cual del mismo modo que el modelo de Servlets centraliza el procesamiento en las páginas de JSP, lo cual implica la integración de código funcional junto con código de presentación.

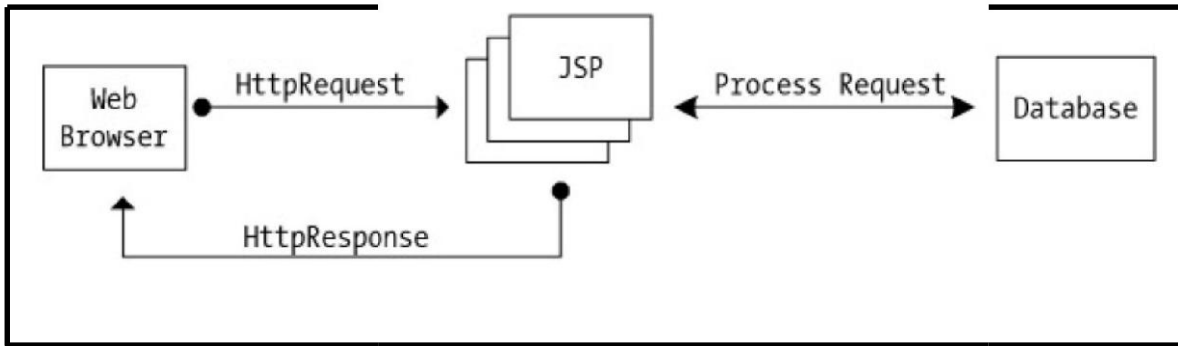


Ilustración 2.10 Modelo 1 de JSP.(Patzer, 2002)

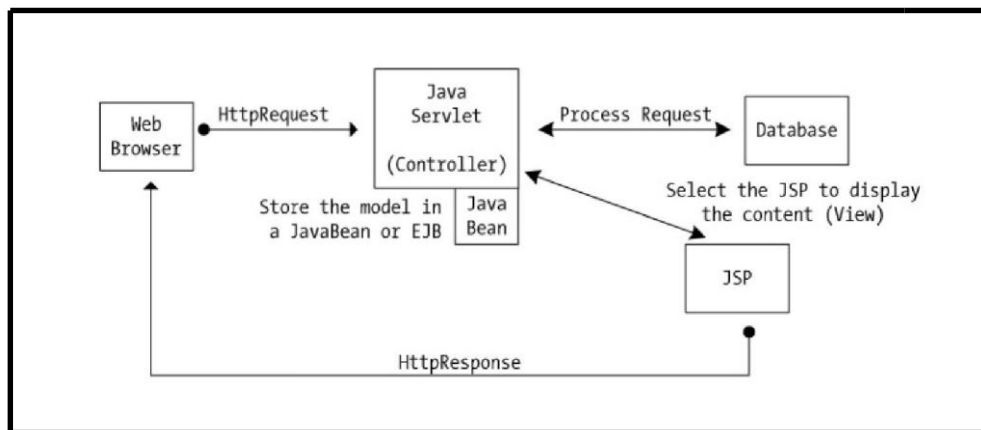


Ilustración 2.11 Modelo JSP 2.(Patzer, 2002)

Para el caso del modelo 2 de JSP se introduce el Modelo Vista Controlador (MVC) el cual separa la lógica de negocios del modelo de base de datos y la interfaz de la aplicación presentando módulos funcionales independientes que al ser integrados coexisten de tal manera que su interacción es imperceptible para el usuario; pero para los desarrolladores brinda facilidades de desarrollo mediante las cuales los equipos de trabajo pueden trabajar de manera separada y ser integradas por un número pequeño de personal lo cual ayuda a una mejor distribución de carga de trabajo para los integrantes del proyecto.



TECNOLÓGICO NACIONAL DE MÉXICO
Instituto Tecnológico de Apizaco

"2015. Año del Generalísimo José-María Moreles y Pavón"



CAPITULO III METODOLOGÍA

METODOLOGÍA PLANTEADA

La naturaleza innovadora de este proyecto dio pie desde un inicio a cambios constantes en su estructura y funcionalidad. Los requerimientos para éste, se fueron adaptando e incrementando según las necesidades del cliente y de acuerdo a un constante análisis de requerimientos además una permanente comunicación con este.

En este capítulo se describe la manera en que se integró metodología planteada para el desarrollo del proyecto, en base a diferentes técnicas las cuales se fueron combinando y adaptando según las necesidades específicas, buscando la realización de un producto que cumpliera completamente con los requerimientos de calidad y funcionales establecidos.

IDENTIFICACIÓN DEL PROBLEMA

Una red inteligente o Smart Grid es un desarrollo integrado de dispositivos de metrología inteligentes; para el caso del presente trabajo, parte del proyecto de la empresa Softek Global eDesign estos dispositivos son los MM. Estos medidores están programados para recibir tareas programables para poder realizar la toma de las lecturas, las cuales son almacenadas por periodos de tiempo relativamente cortos para poder ser enviados cuando sea solicitado a través de la red de medidores. Dentro del GMM se encuentra la CCG, un dispositivo encargado de recolectar los datos producidos por los medidores conectados dentro del gabinete, una vez que ha recibido los datos solicitados esta se encarga de enviarlos por radio frecuencia a través de la red de gabinetes hasta encontrar al coordinador del área donde este se encuentra colocado. El coordinador es el dispositivo que comunicara la red de medidores con el mundo exterior; es decir, envía y recibe los datos de configuración y medición para los medidores que se encuentran bajo su dominio.

Por otro lado tenemos Los sistemas propios de Comisión Federal de Electricidad. Estos sistemas están encargados tanto de la facturación como de la asignación de tareas de medición, altas o bajas de servicio, etc.

Una vez que se han establecido las tareas a realizar, un mensaje es enviado a través de la red de medidores para indicar a estos cuales son los datos que son solicitados o bien las tareas que le son encomendadas. Como resultado, los datos generados son enviados de vuelta hacia los sistemas de CFE

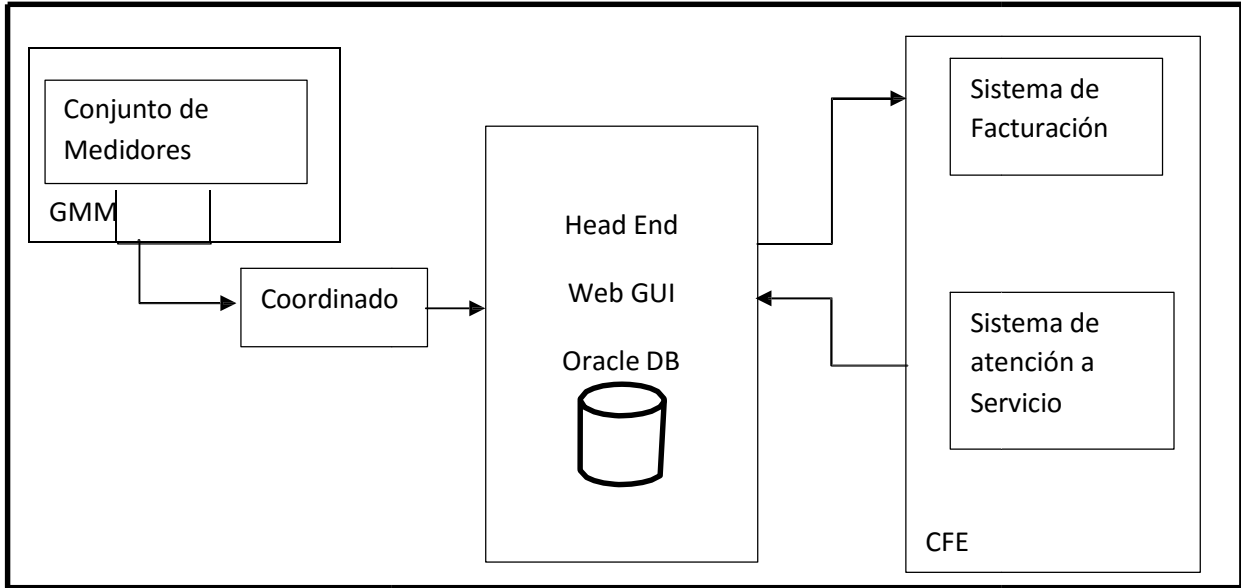


Ilustración 3.1 Diagrama a bloques de la interacción de los diferentes módulos del sistema.

Como podemos observar en la ilustración 3.1 el flujo de información está incompleto sin un sistema que se encargue de comunicar a los sistemas de CFE con la red de medidores y viceversa; este sistema es llamado Head End.

El Head End recibe un mensaje con la asignación de tareas para la red de medidores por parte del sistema de atención a servicios de CFE desde la interfaz web, el cual es interpretado y completado con los registros de la base de datos. Este mensaje entonces ingresa a la red de medidores y es encaminado hasta llegar al medidor específico hacia el cual va dirigido.

Una vez que los medidores han generado la Información solicitada esta es enviada a la CCG, la cual se encarga de mandarla hacia el coordinador que nuevamente se comunica con el Head End, el cual enviara la información solicitada hacia el sistema de facturación de CFE, registrando al mismo tiempo los registros pertinentes en su propia base de datos.

Como podemos observar el Head End tiene como objetivo proveer una interfaz para que el usuario (CFE) pueda interactuar con la red de medidores sin necesidad de presentarse en la locación. Puede brindar a los medidores la información que estos necesitan para poder ejecutar sus tareas de manera automatizada y esta sea recolectada sin intervención directa del Lecturista lo que representa una disminución en el consumo de recursos por parte de la comisión; además de brindar información en tiempo real para que pueda realizar la facturación de sus servicios al igual que desconexiones y reconexiones instantáneas.

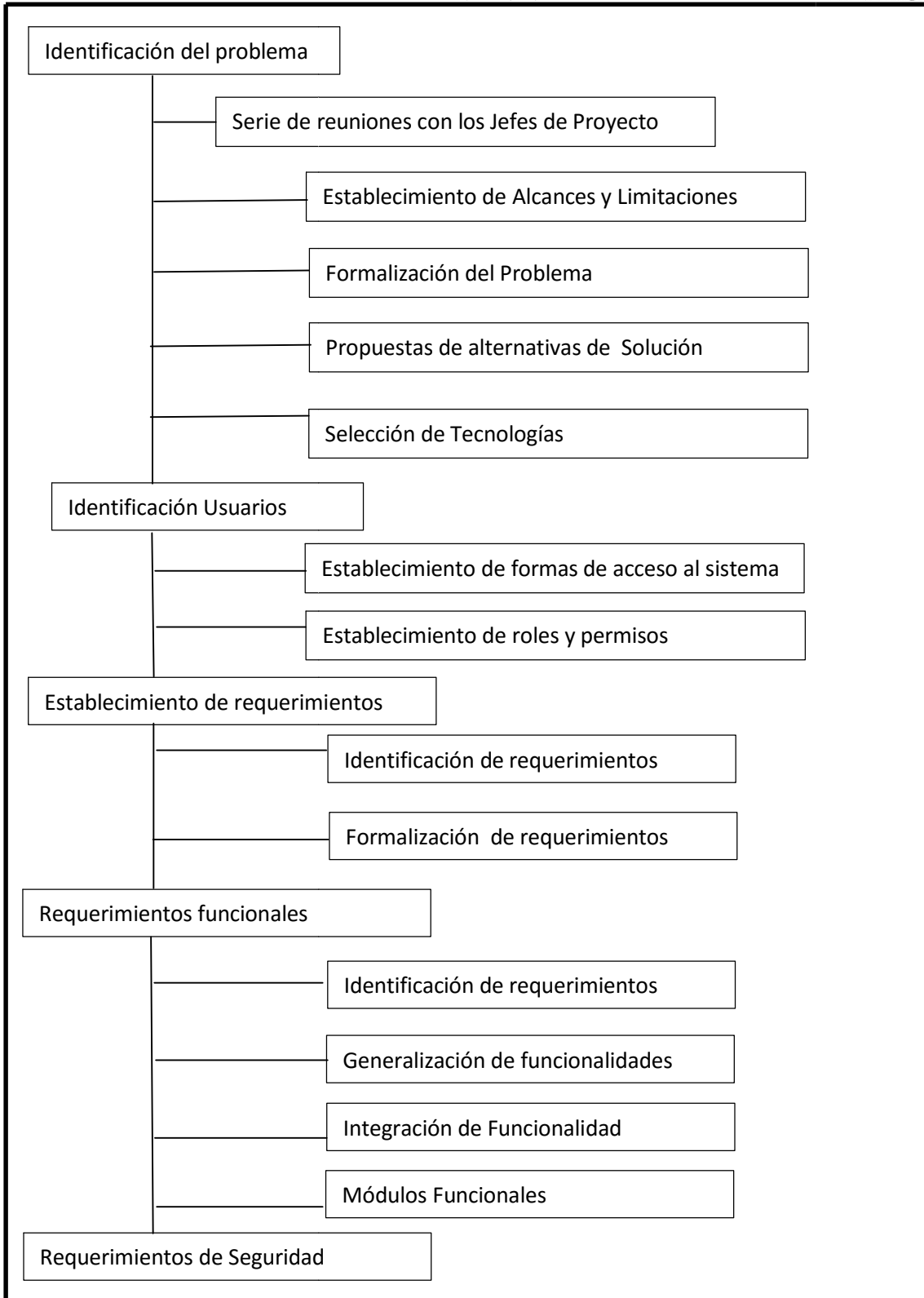


Ilustración 3.2 Diagrama de bloques de la metodología vista de manera general.

ARQUITECTURA DEL SISTEMA.

Basados en el resultado del estudio realizado para el proceso de modelado, análisis y especificación de los requerimientos, así como los obtenidos en la realización de los diagramas ; se plantea el uso una arquitectura de Cliente - Servidor para describir de manera correcta la funcionalidad del sistema Pandora Management System. La arquitectura elegida describe de mejor manera la funcionalidad de PANDORA al presentar los elementos que interactúan en la ejecución de las tareas a través de los diferentes niveles que integran la aplicación y sus medios de comunicación.

La arquitectura de Múltiples Clientes – servidor consiste en algunos clientes que solicitan un servicio, y un servicio que satisface las solicitudes del cliente.

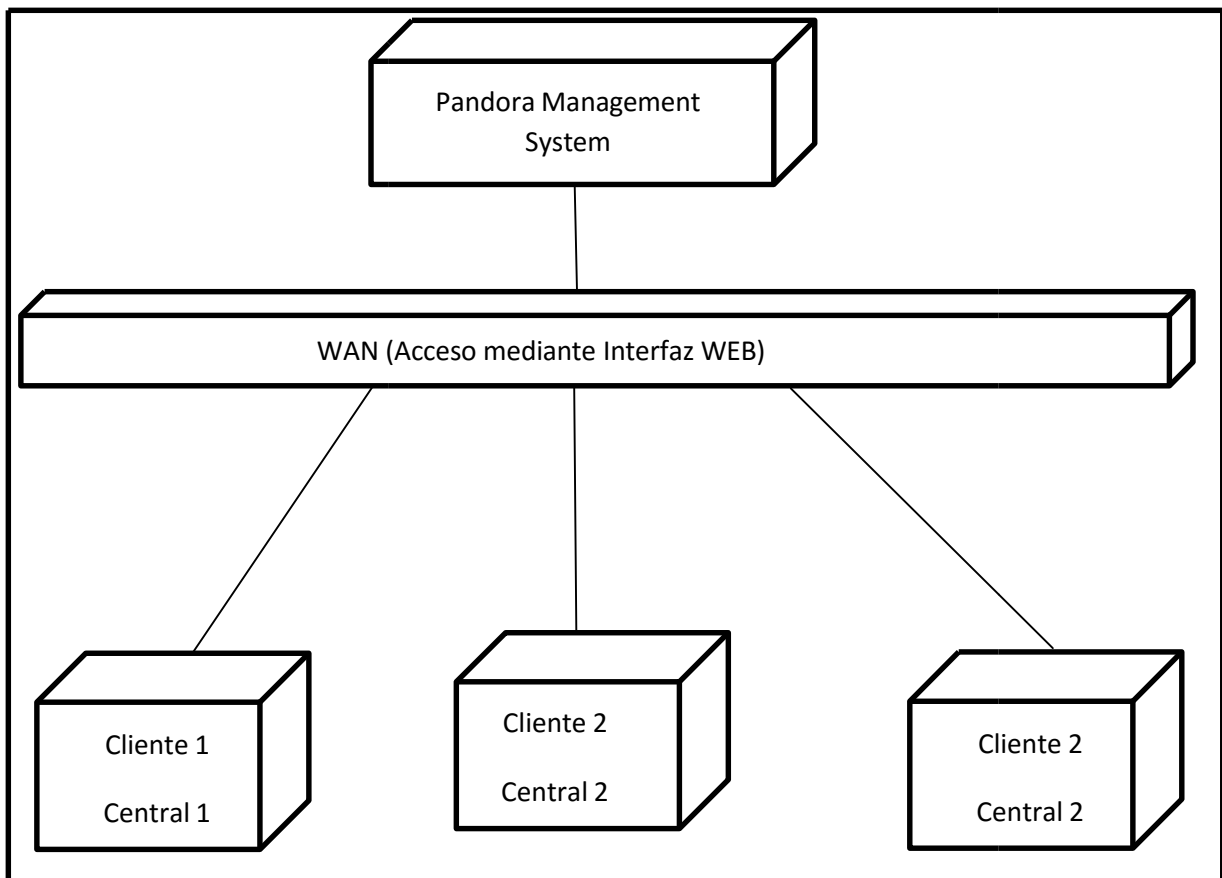


Ilustración 3.3 Patrón de arquitectura de múltiples clientes – Un servidor para PANDORA MANAGEMENT SYSTEM.

En la Ilustración 3.3 podemos observar agrandes rasgos como la interacción descrita en la arquitectura propuesta se realiza por medio de la interfaz web accesada desde internet.

Cada cliente podrá acceder de manera simultánea a la aplicación para poder interactuar con la base de datos o en dado caso generar un archivo XML según la tarea a ejecutar.

De manera más explícita podemos observar como en la ilustración 3.4 se muestran las diferentes interacciones de los módulos que integran el sistema:

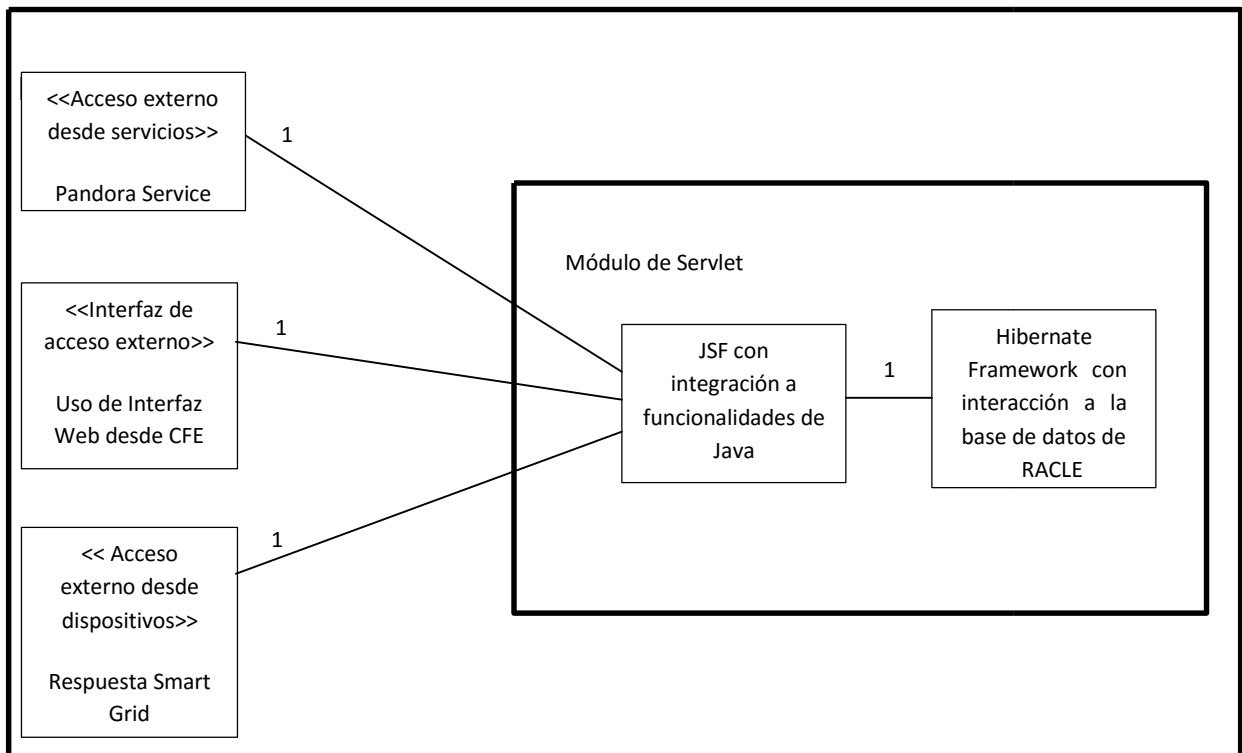


Ilustración 3.4 Diseño de arquitectura múltiples clientes – servidor con descripción de elementos

Podemos observar además los diferentes medios de acceso al sistema como lo son, el pandora Service encargado de la conexión con los Hand Held Devices que interactuara con el Servlet en el núcleo de la aplicación enviando mensajes XML contenedores de la tarea a ejecutar por el sistema, el acceso haciendo uso de la interfaz web desde las terminales en la comisión que habrán de generar tareas y enviaran también archivos XML según la operación realizada, y finalmente una respuesta generada desde el Smart Grid.



TECNOLÓGICO NACIONAL DE MÉXICO
Instituto Tecnológico de Apizaco

"2015. Año del Generalísimo José-María Morelos y Pavón"



El subsistema identificado con el Servlet habrá de interactuar con el módulo de Hibernate, recibiendo los mensajes generados con los archivos XML indicándole la tarea a realizar con los registros de la Base de datos.

REQUERIMIENTOS DEL SISTEMA

Definición:

Sistema integral encargado de la gestión de la información que es producida por la red de medidores así como la información que es enviada hacia la red de medidores.

Objetivo:

Desarrollar un Sistema integral de tipo Head End que se encargue de gestionar la información que es enviada desde el sistema comercial de la Comisión Federal de Electricidad hacia la red de medidores, así como la que es generada por los medidores dentro de la red inteligente en respuesta de las operaciones ejecutadas.

Deberá de recolectar información recabada en campo por los lecturistas por medio de un programa que interactúe con una terminal portátil y lo ponga a disposición de un software para ser trabajado y enviado hacia la red de medidores.

Presentará una interfaz web en la que el trabajador de oficina de CFE podrá acceder y generar tareas programadas para los medidores, así como contendrá funcionalidad para proporcionar gráficas de uso y funcionalidad del propio sistema.

Para este caso el flujo de la información iniciará desde la Utility, que genera un paquete de operaciones dirigidas hacia un MM en específico, este paquete deberá de fluir a través de la Smart Grid entrando por un colector, y pasando por una ruta establecida por el mismo colector de GMM hasta llegar al medidor o medidores hacia los que va dirigido el paquete.

Alcance:

Se desarrollará un daemon basado en un servicio de Windows encargado de la detección automática de la conexión de cualquier terminal portátil, de la cual deberá de extraer un archivo XML contenedor del registro de las operaciones realizadas en campo, el cual haciendo uso de un Web Service deberá de ser enviado hacia el núcleo de la aplicación para ser trabajado.

Se programará la funcionalidad para una interfaz web implementando un Servlet que proporcionará acceso al personal de oficinas de la Comisión a los menús para realizar, altas, edición y eliminación de dispositivos, así como la programación y monitoreo de tareas programadas.



TECNOLÓGICO NACIONAL DE MÉXICO
Instituto Tecnológico de Apizaco

"2015. Año del Generalísimo José-María Moreles y Pavón"



Desarrollo de un Socket para la comunicación del sistema con la red de medidores enviando tablas establecidas bajo el formato XML "MailBox".

No Contempla:

Realización de validaciones para funcionalidad de web GUI (validaciones para cada campo a llenar dentro de la interfaz).

Desarrollo de una funcionalidad para encriptación.

Descripción de Usuarios

Lectorista:

Personal de Comisión Federal de Electricidad encargado de ejecutar las tareas en campo tales como instalación de servicios o toma de lecturas (de ahí el nombre). Interactúa permanentemente con la terminal portátil o Hand Held.

Oficinista:

Personal de la Comisión Federal de Electricidad encargado de las tareas de oficina, además de interactuar con el cliente y el sistema comercial deberá de interactuar con la interfaz web del Head End para programar las operaciones en los dispositivos de la red inteligente.

Reglas de Negocio

La comunicación con la red de medidores estará controlada bajo el protocolo de comunicación ANSI C.12.18 ANSI C.12.19 que define tablas que modelan los objetos para el envío de información a través de la red inteligente.

Todo envío de información a través del sistema integral o Head End o bien deberá de ser encriptado bajo un algoritmo basado en el método de encriptación AES 128.

El acceso a la interfaz web del Head End deberá de ser controlado mediante contraseñas de usuario.

Descripción Detallada de Requerimientos

PANDORA SERVICE

1. Creación para el inicio de un Windows Service
2. Creación de Timer para ciclar la lectura de un directorio
3. Creación de una Función para escribir un archivo CSV desde cero.
4. Creación de una Función para apertura y lectura de un archivo CSV
5. Funcionalidad para transformar el archivo CSV en un arreglo de Cadenas
6. Funcionalidad para obtener valores separados para cada usuario en el CSV

PANDORA

Permitir las siguientes funciones:

1. Lectura de XML
2. Guardar registros en la base de datos.
3. PANTALLA Web para cambios de MRE's, MM, USUARIOS.
4. Generar XML.
5. Generar Log de Operaciones.
6. Adquirir Clases para encriptación.
7. Tabla para lecturas, con fechas y horas y generar archivos /base de datos para almacenar perfiles.
8. Escuchar eventos de manufactura de NUBO.
9. Sincronización con HH (probando con las marcas que tiene CFE y que se utilizaran para realizar las funcionalidades de asociación vía aérea).
10. Asociaciones entre MRE – Usuario- Medidores. Toma de lecturas
12. Cortes: Por usuario / Por gabinete
13. Reconexiones: Por usuario / Por gabinete
14. Alarmas
 - a. Puerta abierta,
 - b. Medidores no autorizados,
 - c. Medidores Huérfanos,

 - d. Falla del medidor 13,
 - e. Temperaturas Max y Min,
 - f. Nivel de la caja de control de PLC / MRE.
 - g. Eeprom Error interna / Externa
 - h. Comunicación con Mrex Fallida
15. Nuevo de MM / MRE
16. Eliminar de MM / MRE
17. Cambiar de MM / MRE
18. Nuevo de Usuario
19. Eliminar de Usuario
20. Eliminar Todos los Usuarios (limpiar Gabinete)
21. Mensajes al MRE.
22. Recepción de confirmación de mensaje leído
23. Mostrar Facturación la MRE
24. Manejo de Finiquitos
25. Registro de operaciones: Por usuario / Por tipo de Operación
26. Expedir reportes en Pantalla de Sincronización con HH
 - a. Asociaciones entre MRE - Usuario- Medidores
 - b. Toma de lecturas

- c. Cortes: Por usuario / Por gabinete
- d. Reconexiones: Por usuario / Por gabinete
- e. Alarmas
- f. Puerta abierta,
- g. Medidores no autorizados,
- h. Medidores Huérfanos,
- i. Falla del medidor 13,
- j. Temperaturas Max y Min,
- k. Nivel de la caja de control de PLC / MRE.
- l. Eprom Error interna / Externa
- m. Comunicación con Mrex Fallida
- n. Reemplazo de dispositivos
- o. Alta de medidores / MRE
- p. Baja de medidores / MRE
- q. Alta de Usuario
- r. Baja de Usuario
- s. Borrar Todo (limpiar Gabinete)
- t. Mensajes al Display: Por usuario / Broadcast
- u. Recepción de confirmación de mensaje leído
- v. Registro de operaciones: Por usuario / Por tipo de operación
- w. Mostrar Facturación al MRE

27. Recibir comandos de Asociaciones entre MRE - Usuario- Medidores

- a. Toma de lecturas
- b. Cortes: Por usuario / Por gabinete
- c. Reconexiones: Por usuario / Por gabinete

Requerimientos No Funcionales

● Requisitos de rendimiento

El sistema deberá de ser capaz de ejecutarse de manera simultánea en red, es decir deberá de permitir múltiples accesos al mismo tiempo desde lugares distribuidos geográficamente, brindando fluidez en la ejecución e integridad en la información que se presenta al cliente.

Consecuentemente se selecciona el desarrollo de la interfaz web basada en la tecnología de Servlet de Java; en la cual la carga de ejecución radica del lado del servidor y no del cliente decrementando los requisitos mínimos para ejecución a simplemente un ordenador con conexión a internet.

● Seguridad

Dentro de las interacciones del sistema, deberá de garantizarse la integridad de la información, garantizando que esta es enviada de manera segura por entre los módulos del sistema.

Por lo tanto se implementará un algoritmo de encriptación basado en AES 128 que garantizará la confidencialidad de la información al ser transportada.

● Disponibilidad

EL sistema deberá de estar disponible en todo momento para poder realizar la ejecución de tareas establecidas anteriormente, por lo cual el sistema deberá ser implementado en un servidor remoto al cual se ingresará por medio de una URL teniendo de este modo acceso al Servlet.

● Mantenibilidad

Se eligió una arquitectura de cliente servidor para el desarrollo de la aplicación donde se establecen módulos para cada parte de la aplicación que deberá de ejecutarse de una manera más independiente, aunque todas interactúen constantemente.

Las ventajas aportadas por esta arquitectura incluyen un desarrollo incremental, es decir en caso de ser requerida una nueva funcionalidad, esta se agrega como una nueva clase enviando y recibiendo los parámetros necesarios sin necesidad de mayores cambios.

● Portabilidad

Al ser una aplicación que se ejecuta desde el servidor donde esta se encuentra cargada, se puede acceder desde prácticamente desde cualquier dispositivo con una conexión a internet.

Requerimientos de Documentación de Usuario y Sistemas de Ayuda

- Manual de Usuario.
- Documentación de código especificada en los códigos preparada para la herramienta Doxygen.

Restricciones de Diseño

El sistema deberá de interactuar sin problemas con las aplicaciones desarrolladas para la terminal programada en C#, y los web services de NUBO programados en JAVA.

Interfaces

Interfaces de Usuario:

Interfaz Web la cual será diseñada implementando el framework Prime Faces para java, el cual brinda una serie de componentes prediseñados visualmente atractivos y funcionalmente intuitivos para facilitar la interacción con el sistema.

Requerimientos Funcionales

Entradas:

Para esta sección la identificación de usuario se realizará mediante un usuario y contraseña.

Procesos:

Todo proceso estará directamente relacionado con el núcleo del sistema desarrollado en el lenguaje de programación java que cuenta con la interfaz web.

A través de esta el usuario es capaz de realizar las operaciones requeridas ingresando los datos requeridos para cada caso y obteniendo la información específica generada como resultado de la ejecución de sus operaciones.

Salidas:

El usuario será capaz de poder observar el resultado de la ejecución de las operaciones, además de estadísticas de uso del producto a través de la sección de reportes dentro del mismo.

En caso de cualquier error de ejecución, el usuario debe ser informado al momento para tomar las acciones necesarias.

Selección de metodologías para diseño y planeación.

La primera metodología que sirvió como guía en el proceso de desarrollo es la metodología ágil de Scrum, seleccionada por su naturaleza incremental y adaptable.

Los roles establecidos por la mencionada metodología estaban cubiertos por:

- Product Owner: Jefes de proyecto y gerencia de la empresa tecnologías EOS.
- Scrum Master: Jefe de proyecto de la empresa Softek Global eDesign.
- Developer Team: Jefe de proyecto de Softek, autor de este documento. Cabe mencionar que debido a las dimensiones del desarrollo, se agregaron nuevos miembros al equipo de desarrollo, los cuales fueron integrados cerca del final de la fase de diseño cumpliendo con una participación significativa en la fase de desarrollo de la interfaz gráfica web; la cual será descrita a detalle posteriormente en este documento.

Según las bases de ingeniería de software el primer paso realizado, una vez que el equipo de desarrollo fue establecido, fue el análisis de requerimientos, el cual

mediante constantes reuniones con el cliente, arrojó como resultado un documento de especificación de requerimientos el cual según los resultados alcanzados en cada entrega se iba complementando, asignando prioridades a estos, desechando los que estaban mal planteados y agregando las nuevas funcionalidades encontradas.

Este documento fue colocado en línea y se brindó acceso a todas las personas que se encontraban dentro del equipo de desarrollo al igual que a la gerencia de la empresa para que esta estuviera al tanto de los objetivos alcanzados y el porcentaje de avance del proyecto en cualquier momento. A los desarrolladores se les asignó privilegios de edición, por lo cual podían agregar nuevas funcionalidades según su juicio, las cuales eran evaluadas posteriormente por el equipo para su aprobación y calendarización.

En la Ilustración 3.5 podemos observar un diagrama de bloques de la funcionalidad del Head End de manera muy abstracta describiendo las funciones más básicas para el sistema.

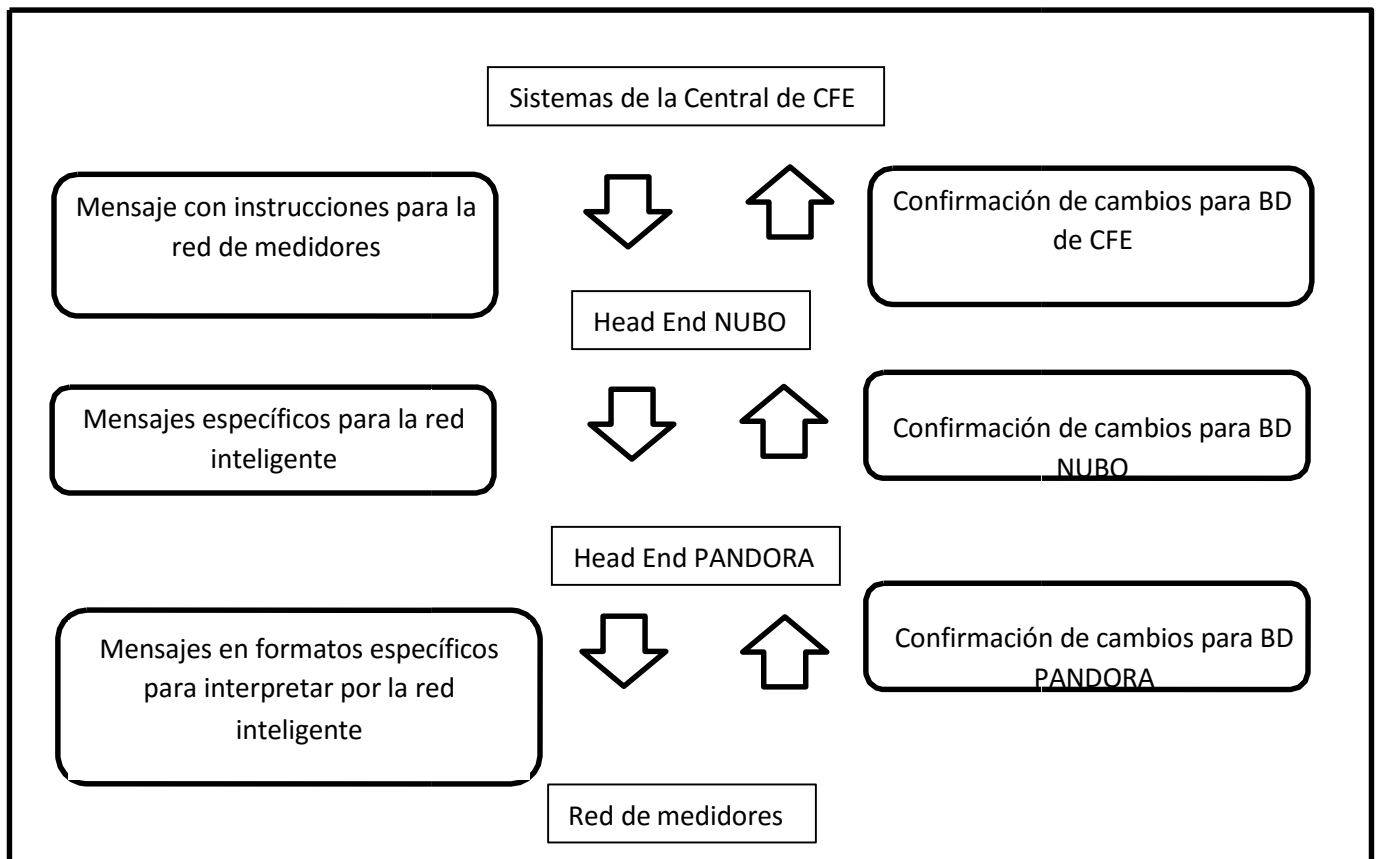


Ilustración 3.5 Diagrama a bloques de funcionalidad generalizada.

Para el proceso de recepción de información el sistema desarrollado recibe una lista de tareas para ser enviadas hacia la red de medidores que es creada desde los sistemas informáticos propiedad de CFE, esta lista de tareas contiene mensajes para todas las marcas proveedoras de medidores analógicos o digitales; NUBO, al ser el único Head End que puede establecer comunicación directa con los sistemas de la comisión, recibe una lista completa de tareas y medidores la cual segmenta por tareas de acuerdo a las marcas de medidores dentro de la red y las envía hacia las redes de medidores específicas; es entonces cuando Pandora Management System (PMS) entra en funcionamiento, este Head End Propiedad de EOS – Softek recibe la lista de tareas asignadas para los medidores de la red inteligente descrita previamente en este documento, las tareas mencionadas pueden ser por ejemplo:

- Solicitud para la toma de lecturas en tiempo real.
- Desconexión de dispositivos.
- Reconexión de dispositivos.

Una vez que el Head End recibe el mensaje con la lista de tareas con número de medidor asignado, este va en busca de información complementaria en su base de datos para que el mensaje pueda ser enviado y direccionado de manera correcta hasta que este alcance su destino dentro del segmento de red apropiado.

Una vez recibido el mensaje por el medidor correcto, se generará una respuesta que se envía a través de la red de medidores hasta alcanzar al dispositivo colector. El Colector envía directamente el mensaje al Head End mediante una conexión LAN Ethernet y en el momento que es recibido el mensaje por PMS, se registran los cambios nuevamente en la base de datos para poder ser enviados a NUBO y este informe a los sistemas de CFE acerca de los cambios realizados hasta que finalmente todos los cambios realizados son registrados junto a su estatus, en caso de ser efectivos son registrados; para el caso contrario, se programa una serie de reintentos.

SELECCIÓN DE LOS LENGUAJES DE PROGRAMACIÓN.

Al haber terminado la fase de diseño y planeación se comienza con el desarrollo del sistema, por lo cual se realizó una selección de lenguajes de programación bajo un concepto crucial: compatibilidad. El desarrollo del Head End fue completamente desde cero, dejando en las manos del equipo desarrollador la selección de las tecnologías más óptimas para desarrollar un producto de calidad y funcional, pero debido a que el sistema debe interactuar el equipo se decidió por JAVA y C#, ambos lenguajes de programación orientados a objetos.

En primera instancia tenemos las terminales portátiles o Hand Held (HH), una vez que el agente de campo o lectorista vuelve de realizar sus operaciones conecta la HH a una terminal USB llamada “cuna” para descargar los registros de actividades realizadas.

Debido a que la aplicación de la HH fue programada en C# se optó por desarrollar la aplicación encargada de descargar la información proveniente de esta en el mismo lenguaje de programación, y de este modo evitar problemas de compatibilidad.

Debido a que este proceso de recolección debe ser ejecutado de manera automática se decidió por realizar un Daemon que funciona en segundo plano registrando los eventos realizados en un Log Registry para que en caso de algún mal funcionamiento esto sea informado al Head End y se tomen las medidas necesarias y del mismo modo se informe al usuario en tiempo real de lo acontecido.

SELECCIÓN DE LA BASE DE DATOS.

Para la selección de la base de datos se realizó un estudio comparativo de los dos manejadores con más renombre dentro del campo, MySQL y ORACLE; con la finalidad de poder establecer el más confiable y capaz de almacenar la información del sistema.

Oracle es un manejador de base de datos ORDBMS (Object-Relational Data Base Management System) multiplataforma desarrollado por Oracle Corporation.

Tabla 3.1 Características de las distintas versiones de Oracle (Precios de 2015).

	Express Edition	Estándar Edition One	Standard Edition	Enterprise Edition
Precios:	Gratis	\$2,645.00 - \$85,175.00	\$5,140.00 - \$256,993.00	\$69,750.00 - \$697,552.00
Máximo	1CPU	2 Sockets	4 Sockets	Sin Limites
RAM	1GB	Max SO	Max SO	Max SO
Tamaño de la base de datos	11GB	Sin Limite	Sin Limite	Sin Limite

Como se puede observar en la tabla 3.1 existe una gran diferencia en capacidades y precios entre las distintas versiones de las bases de datos de Oracle, aun así todas las versiones presentan confiabilidad y potencia, la versión más restringida es la gratuita al tener un límite de 11 GB de almacenamiento para los registros.

Cabe resaltar que a pesar de tener esta limitante de almacenamiento la versión gratuita representa una opción completamente viable y confiable para pequeñas empresas y para el caso del presente desarrollo para pruebas piloto.

My SQL es un sistema de gestión de bases de datos relacionales multiplataforma desarrollado por Sun Microsystems (Propiedad de Oracle).

Tabla 3.2 Características de las distintas versiones de MySQL (precios de 2015).

	Standard Edition	Enterprise Edition	Cluster Grade Edition	Carrier Edition
Suscripción anual	USD 2000	USD 5000	USD 10000	
Soporte 24/7	√	√	√	
Base de conocimientos	√	√	√	
Mantenimiento	√	√	√	
Servidor de Base de Datos	√	√	√	
Conectores	√	√	√	
Replicación	√	√	√	
particionamiento		√	√	
Utilidades		√	√	
Workbench	√	√	√	

En la tabla 3.2 se muestra una comparativa entre los productos de MySQL con costo; las opciones más poderosas de SQL las cuales podrían competir con Oracle, del mismo modo que con las versiones de Oracle las versiones más pequeñas se encuentran más restringidas en cuanto a funcionalidad, y características.

Por otra parte en la Tabla 3.3 se presenta una comparativa funcional entre los dos manejadores a detalle.

Tomando en cuenta los detalles presentados en el estudio anterior se optó por usar la versión Oracle Express 11g para la primera fase de despliegue de la red inteligente debido a su robustez, confiabilidad y seguridad, y se planea migrar a versiones de paga conforme vaya creciendo el número de medidores instalados en campo y por consecuente la cantidad de registros dentro de base de datos.

Tabla 3.3 My SQL versus ORACLE características / Funcionalidad

Nota: George Trujillo (May 28, 2008) "MySQL versus Oracle Features/Functionality"

Recuperado de: https://blogs.oracle.com/GeorgeTrujillo/entry/mysql_versus_oracle_features_functionality

Características / Funcionalidades	MySQL	Oracle
Fortalezas	Precio / Desempeño Gran desempeño cuando las aplicaciones toman ventaja de la arquitectura	Base de datos Aircraft carrier capaz de correr OLTP y VLDBs grandes.
Productos de Base de Datos	Enterprise (Costo bajo) – con soporte, más estable. Community (gratis) – más actual.	Enterprise (Costo alto) Standard (Costo medio) Standard One (Costo bajo) Express (gratis) – hasta 11GB
Perspectiva de aplicación	Las aplicaciones web a menudo no sacan ventaja de la funcionalidad del servidor de base de datos. Las aplicaciones web se enfocan más en una lectura rápida.	Entre más hagas en la base de datos más amaras Oracle con compilaciones PL/SQL, XML, APEX, Java, etc.
Administración	Puede llegar a ser trivial al momento de configurarlo y correrlo. Las configuraciones grandes y avanzadas se pueden volver complejas	Requiere de conocimiento profundo y habilidades para manejar entornos grandes. Puede llegar a ser extremadamente complejo pero también muy poderoso.
Popularidad	Extremadamente popular con compañías web, principiantes,	Extremadamente popular en Fortune 100, Aplicaciones para

	negocios pequeños o medianos, proyectos pequeños y medianos.	negocios de empresas medianas y grandes y almacenes de datos medianos y grandes.
Dominios de aplicación (Mas populares)	Web excels) (MySQL Data Warehouse Video Juegos Ambientes pequeños y medianos. OLTP y	Aplicaciones empresariales OLTP grandes y medianas. Oracle tiene excelencia en aplicaciones para grandes aplicaciones de negocios (EBS, Siebel, PeopleSoft, JD Edwards, Retek, ...) Almacenamiento de datos mediano / grande.
Ambientes de desarrollo (Más comunes)	1) PHP 2) Java 3) Ruby on Rails 4) .NET 5) Perl	1) Java 2) .NET 3) APEX 4) Ruby on Rails 5) PHP Nota: Oracle se enfoca en java para las aplicaciones de negocios de nueva generación.
Servidor de Base de Datos (Instancia)	La instancia de la base de datos almacena memoria global en el proceso de Segundo plano mysqld Las sesiones de usuario se manejan con hilos.	La instancia de base de da tos tiene numerosos procesos de Segundo plano dependientes de la configuración. El área global del sistema es memoria compartida para SMON, PMON, DBWR, LGWR, ARCH, RECO, etc. Las sesiones se

		manejan a través de procesos del servidor
Servidor de Base de Datos (almacenamiento físico)	<p>Creación de esquemas de base de datos</p> <p>Cada motor de almacenamiento guarda información de manera diferente</p> <p>Motores de almacenamiento más comunes:</p> <p>MYISAM – almacena datos en .FRM, .MYD y archivos .MYI</p> <p>InnoDB – almacena datos en un espacio de tablas común o individual por tabla.</p> <p>Registros binarios son usados para una restauración de “un punto en el tiempo”</p>	<p>Utiliza tablespaces para los metadatos del Sistema, datos de usuario e índices.</p> <p>Los tablespaces más comunes incluyen:</p> <p>SYSTEM</p> <p>SYSAUX</p> <p>USER DATA</p> <p>USER INDEXES</p> <p>TEMPORARY</p> <p>UNDO</p> <p>Redo y archivos de registro son usados para restauración de un punto en el tiempo.</p>
Tablas	<p>Las tablas usan motores de almacenamiento.</p> <p>Cada motor proporciona diferentes características y comportamiento.</p>	<p>Pocas tablas con demasiadas características.</p>
Particionamiento	<p>Gratis,</p> <p>Características básicas</p>	<p>Costo alto con demasiadas opciones</p>
Replicación	<p>Gratis,</p>	<p>Costo alto, demasiadas</p>

	relativamente fácil de configurar y manejar. Características básicas pero funcionales. Gran escalabilidad horizontal.	características y opciones. Mucho más grande en complejidad con un muchas características. Permite muchas maneras de filtrar y manipular datos.
Transacciones	InnoDB y futura integración de los motores de almacenamiento Falcon y Maria.	Las tablas regulares de solo índices soportan transacciones.
Respaldo/Recuperación	No existe respaldo en línea. Replicación Capturas de SO Respaldo rápido InnoDB	El manejador de restauraciones (RMAN) soporta respaldos rápidos y corre como un repositorio de central separada para múltiples servidores de Oracle.
Exportar/Importar	Fácil, Muy básico	Mas características
Diccionario de Datos (catalogo)	Information_schema y los esquemas de base de datos mysql ofrecen metadatos básicos	El diccionario de datos ofrece mucha información detallada para el mejoramiento. Oracle comienza a cambiar para el uso de estructuras de metadatos.
Manejamiento/Monitoreo	Costo bajo, el MySQL Enterprise Monitor ofrece funcionalidad básica. Soluciones adicionales de código libre Debe usar además Scripts de	Costo alto, el Grid Control ofrece una gran variedad de funcionalidades. Una gran cantidad de opciones de terceros tales como BMC, Quest, Embarcadero y CA.

administrador		
Almacenamiento	Cada motor de almacenamiento usa diferente almacenamiento. Variando según los archivos individuales o tablespaces.	Las tablas son manejadas en tablespaces, ASM ofrece particionamiento y respaldo de almacenamiento usando discos rápidos y baratos.
Stored Procedures	Funciones muy básicas, corre interpretado en hilos de sesión. Escalabilidad limitada.	Características avanzadas, corre interpretado o compilado. Gran cantidad de paquetes diseñados con funcionalidad significativa. Extremadamente escalable.

DISEÑO DE LA BASE DE DATOS

La base de datos tiene como principal funcionalidad guardar un registro de las actividades realizadas por la red de medidores así como las dependencias de los dispositivos para con los usuarios.

Cabe resaltar que CFE cuenta con sus propias bases de datos donde guarda la información propia para que realice sus procesos de facturación, pero debido a la naturaleza innovadora del proyecto se encontró la necesidad de crear una base de datos donde se ubicaran las asociaciones entre dispositivos; por ejemplo los medidores que están relacionados con un gabinete, y de todos los dispositivos hacia el usuario el cual corresponde el servicio.

Esta base de datos fue cambiando a lo largo del desarrollo del proyecto agregando tablas, eliminando campos y alterando a estos mismos debido al crecimiento y mejoramiento continuo del proyecto

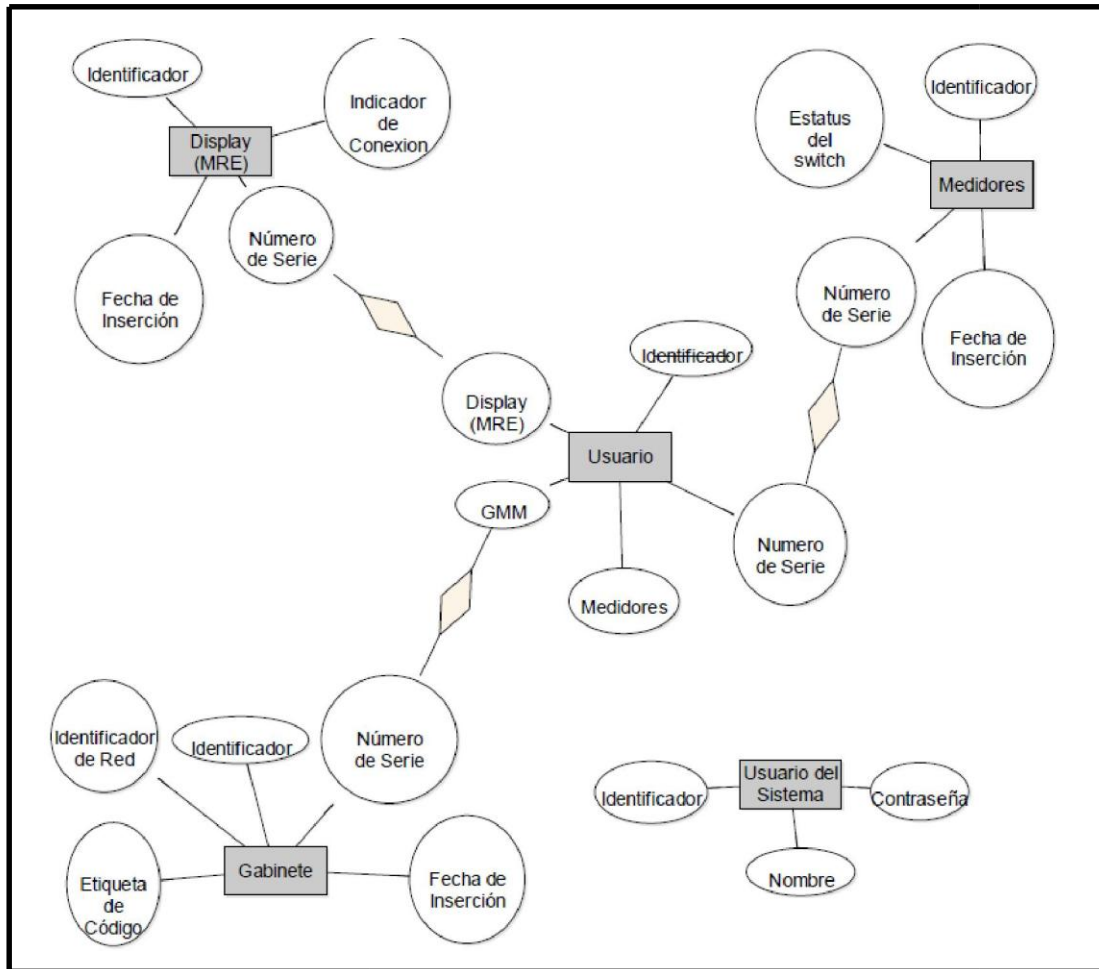


Ilustración 3.6 Diagrama Entidad – Relación Utilizado para el diseño de la base de datos.

Tras haber realizado el análisis de requerimientos por el equipo desarrollador se modeló un diagrama de Entidad – Relación el cual se muestra en la Figura 3.6.

La tabla principal es la tabla de usuarios hacia la cual se dirigen todas las relaciones. Cada dispositivo se debe de relacionar con un usuario, esto quiere decir que en la instalación física son los dispositivos que corresponden a un servicio dado.

Cada dispositivo descrito cuenta con un identificador propio de la base de datos para poder asignar un orden de inserción a los registros que son salvados dentro de esta. Así como un número de serie el cual lo identifica de todos los otros dispositivos que son agregados dentro de la red. El campo fecha de inserción se refiere a la fecha en la cual el dispositivo fue agregado a la red inteligente de medidores, lo cual nos sirve para poder presentar estadísticas de uso y tiempo de vida de los dispositivos.

Los medidores cuentan con un campo el cual nos permite saber el estado de su conexión al que llamamos estatus del Switch; esto quiere decir, nos permite saber si está activo o no, tomando lecturas de un servicio funcional.

Los módulos remotos de energía presentan un campo mediante el cual permitirán al sistema saber su estado de conexión, esto quiere decir si es que esta enlazado a los medidores propios del servicio o no.

Los gabinetes por su parte cuentan con una etiqueta que los identifica, esta es diferente a su número de serie y comienza con la letra Z, además, esta tabla cuenta con un campo que permite al sistema saber el identificador dentro del segmento de red que tiene asignado el gabinete en este momento facilitando de este modo el envío de paquetes hacia los dispositivos dentro de la red.

Tabla 3.4 Diseño conceptual de la tabla GMM

Columna	Tipo de Datos
ID	NUMBER
SERIAL_NUMBER	VARCHAR
Z_CODE	VARCHAR
ISERTION_DATE	DATE
LAN_ID	VARCHAR

Para el diseño de la tabla de los gabinetes se plantea el esquema presentado en la Tabla 3.4 con los tipos de datos Number para el ID como auto incrementable, el número de serie como un Varchar, para la etiqueta de código Z un tipo de dato Varchar, para el registro de la fecha en la que el dispositivo fue integrado en la red se planteó un campo Date y finalmente para el identificador del segmento de red se incluyó un campo de tipo Varchar.

Tabla 3.5 Diseño conceptual de la tabla MM

Columna	Tipo de Datos
ID	NUMBER
SERIAL_NUMBER	VARCHAR
SWITCH ESTATUS	NUMBER
ISERTION_DATE	DATE

Los tipos de datos especificados para la Tabla 3.5 de Medidores son, Number para el identificador dentro de la base de datos que es auto incrementable, Varchar para el número de serie propio de cada dispositivo, cabe resaltar que se puede agregar de

uno hasta tres medidores según el tipo de servicio al que pertenezca (monofásico, bifásico, trifásico). El estatus del Switch, campo que nos indica si el dispositivo está disponible para servicio o no es de tipo Number representado por un estatus binario según el estado del dispositivo al momento. La fecha de inserción del dispositivo a la red inteligente se registra en un campo con tipo de datos Date.

Tabla 3.6 Diseño conceptual de la tabla MRE

Columna	Tipo de Datos
ID	NUMBER
SERIAL_NUMBER	VARCHAR
LINKESTATUS	NUMBER
ISERTION_DATE	DATE

El dispositivo MRE el cual presenta una estimado del consumo en tiempo real al usuario se modela en la Tabla 3.6 que cuenta con un campo de identificador auto incremental de tipo Number, un campo Varchar para el número de serie propio del dispositivo, el campo Link Status que representa el estado de conexión del dispositivo (si se comunica con el gabinete de medidores), es de tipo Number haciendo la función de un estado binario para verdadero o false según sea el caso.

Finalmente el campo Insertion Date nos presenta la fecha en que el dispositivo fue agregado a la red de medidores en un campo de tipo Date.

Tabla 3.7 Diseño conceptual de la tabla PANDORA USERS

Columna	Tipo de Datos
ID	NUMBER
USER_NAME	VARCHAR
PASSWORD	VARCHAR

Los usuarios del sistema se presentan en una tabla llamada "Pandora Users" la cual contiene un identificador propio de la base de datos de tipo Number, el nombre de usuario se presenta en un campo de tipo Varchar al igual que la contraseña.

La tabla 3.8 de usuarios, a la cual estarán relacionadas todas las tablas de dispositivos para representar la asociación de estos con el servicio cuenta con su propio campo de identificador de tipo Number, un número de serie o RPU (identificador del servicio para CFE) de tipo Varchar, tres campos para poder agregar los medidores correspondientes al tipo de servicio, el campo MM2 y MM3 pueden ser nullos. Incluye también campos de tipo Varchar para relacionar los identificadores del MRE y el Gabinete de medidores al cual están vinculados los dispositivos.

Tabla 3.8 Diseño conceptual de la tabla USER

Columna	Tipo de Datos
ID	NUMBER
SERIAL_NUMBER	VARCHAR
MM1	VARCHAR
MM2	VARCHAR
MM3	VARCHAR
MRE	VARCHAR
GMM	VARCHAR
INSERTION_DATE	DATE

El campo Insertion Date de tipo Date nos muestra la fecha en que fue dado de alta del servicio.

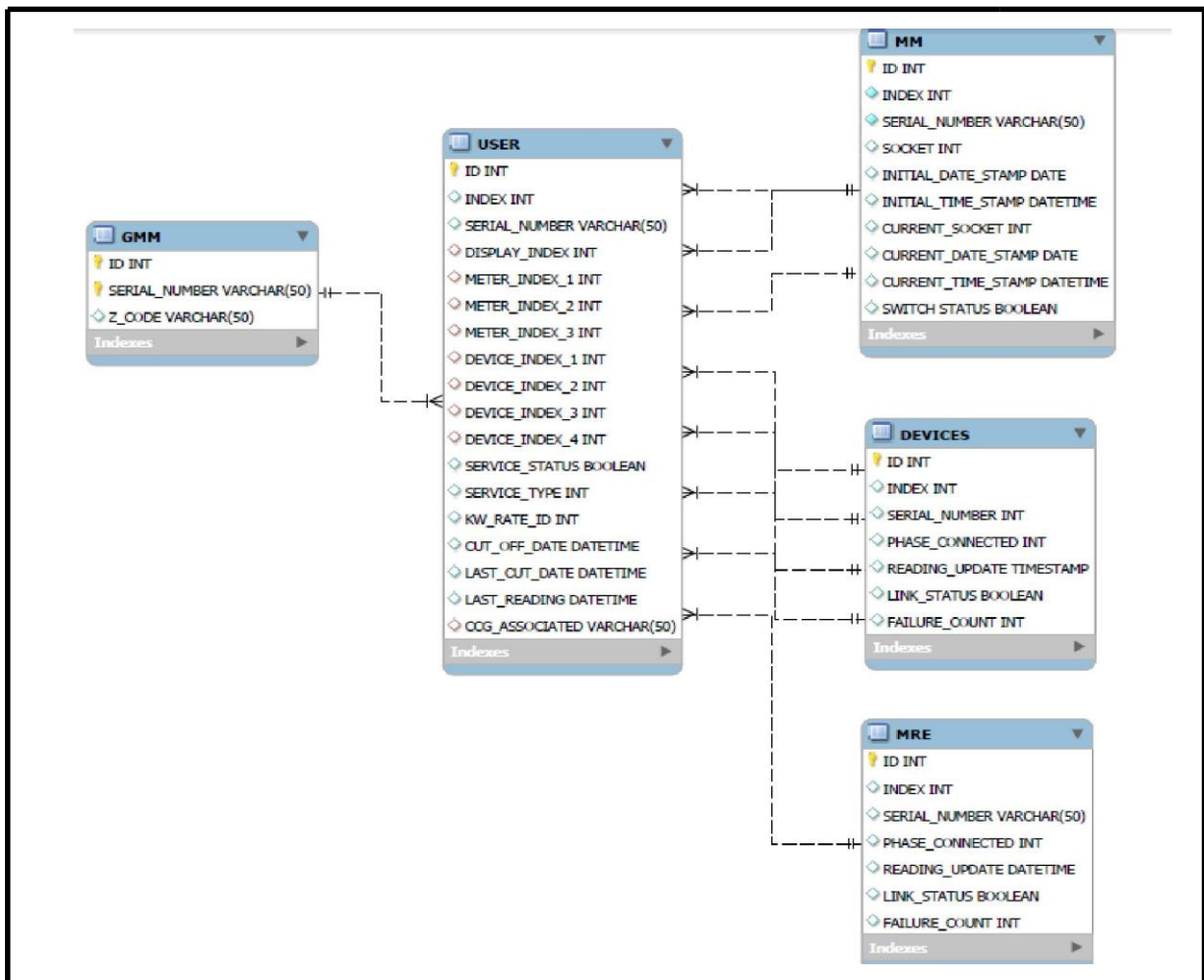


Ilustración 3.7 Diseño de la base de datos.

En la Ilustración 3.7 se pueden observar las relaciones entre las entidades que constituyen las tablas dentro de la base de datos para el Head End.

Como se mencionó anteriormente uno de los objetivos primordiales para el sistema es el de mantener un registro de las asociaciones de dispositivos con el usuario respectivo por lo cual uno de los fundamentos principales en la etapa de diseño y arquitectura para la base de datos fue crear tablas donde se registrarán los campos indispensables para mantener en orden las asociaciones y mantener un tamaño relativamente pequeño de los registros en el servidor sin descuidar la integridad de los datos. La base de datos fue adaptándose del mismo modo que el resto del sistema conforme los requerimientos iban evolucionando, así que nuevas tablas fueron agregadas, campos fueron eliminados y relaciones fueron modificadas; pero la funcionalidad siempre fue capaz de integrarse de manera satisfactoria gracias a las metodologías implementadas para el desarrollo.

DIAGRAMAS DE CASOS DE USO.

Los diagramas de casos de uso describen las interacciones del sistema con los usuarios finales de este, así como el flujo de tareas y datos para que el desarrollador tenga una idea en un nivel diferente de abstracción de cómo es que funcionará el Head End en términos generales.

La funcionalidad del sistema de generalizada es servir como puente de comunicación entre la red de medidores y la comisión federal de electricidad. Para este diagrama se emite la integración con el sistema NUBO debido a que funciona como una indispensable puerta de entrada para todo sistema que deba establecer comunicación con los sistemas informáticos de la comisión. Al Head End descrito en este documento le facilita una lista de tareas exclusiva de los dispositivos que integran la red inteligente de Softek Global. Como se puede observar en la figura 3.8 tenemos a los dos actores principales del sistema, el oficinista y el lectorista. Ambos pueden interactuar con el núcleo funcional del sistema mediante el acceso a dos módulos principales, Pandora Service y Pandora WEB GUI.

Como se puede observar en la Ilustración 3.9 un nuevo actor es integrado; este es el Web Service, este servicio es integrado solo en algunas secciones del sistema y tiene una funcionalidad limitada a transferencia de archivos, por lo cual no es considerado como un actor general en la funcionalidad completa de Pandora.

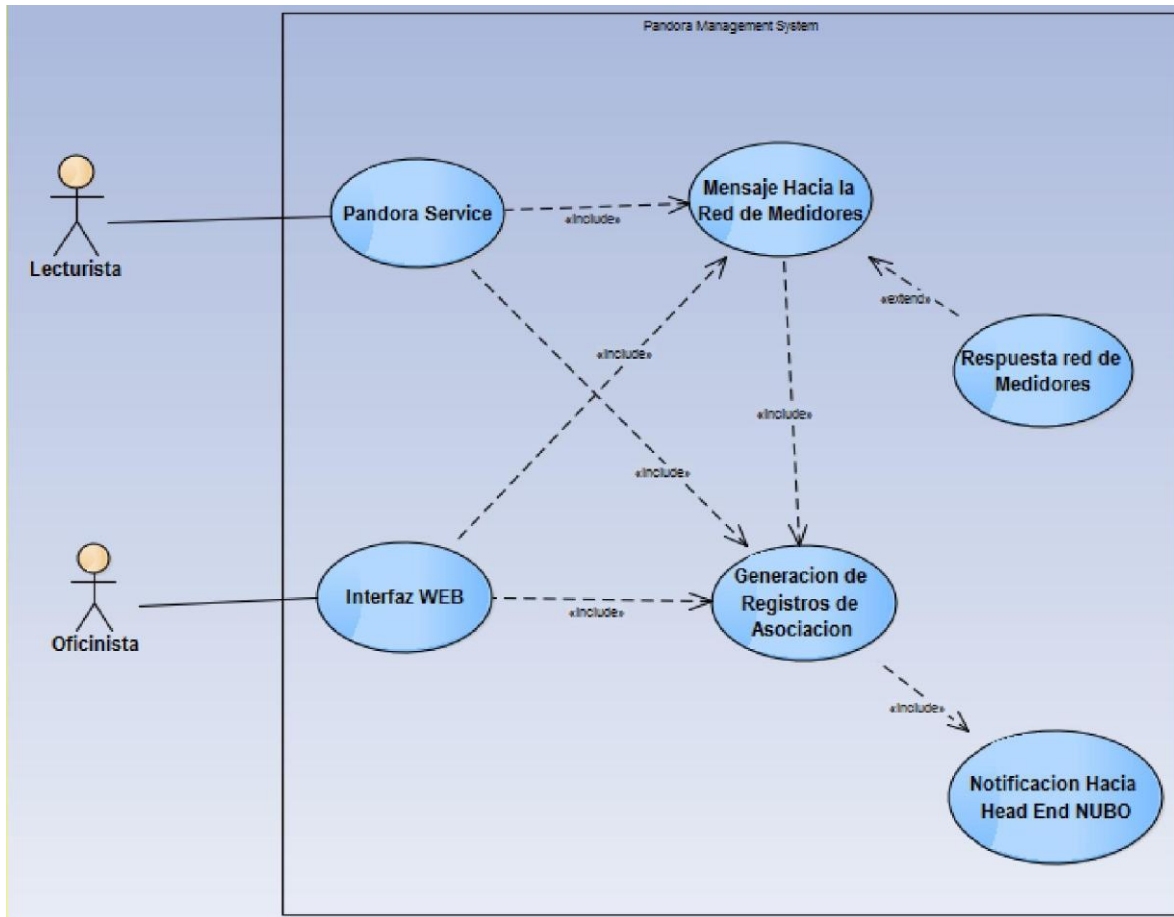


Ilustración 3.8 Diagrama de caso de uso para la funcionalidad generalizada del sistema.

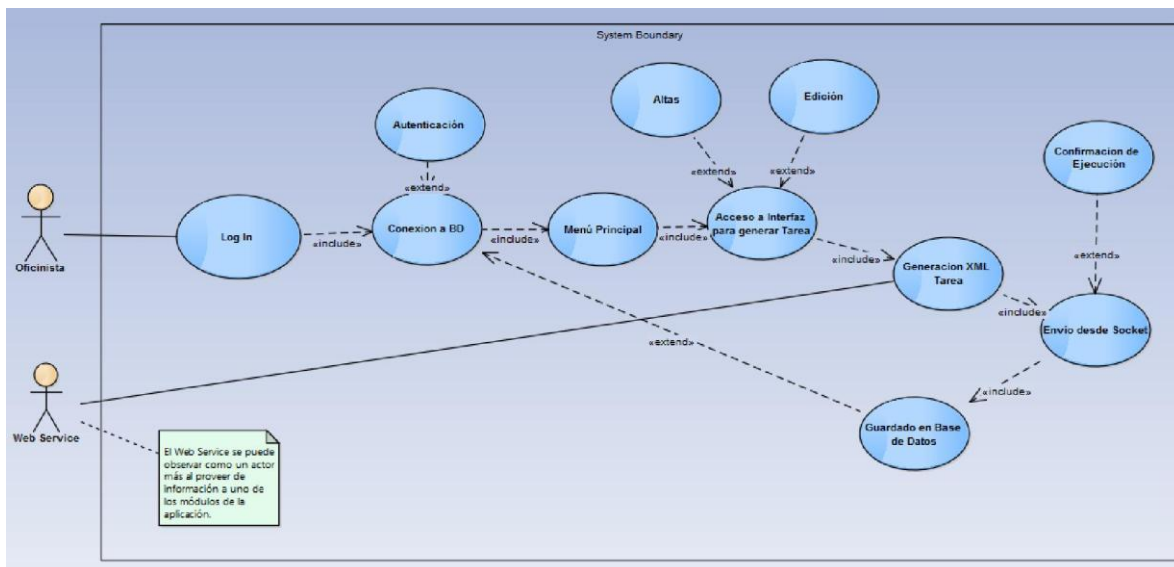


Ilustración 3.9 Diagrama de funcionalidad general para el módulo Web.

Para el módulo Web se describen dos maneras de interacción, accedendo como oficinista o bien trabajando en conjunto con un Web Service provisto por NUBO. El oficinista ingresa al sistema desde la pantalla de Log In donde ingresa sus datos que serán cotejados con la base de datos para poder darle acceso. Una vez realizada la autenticación del usuario, se le muestra el menú principal mediante el cual tendrá control sobre todas las operaciones que se pueden realizar para servicios y dispositivos cotejados con la base de datos de Oracle; estas operaciones pueden ser por ejemplo altas, bajas y ediciones de registros los cuales serán reflejados en la propia base de datos de Pandora y la base de datos de CFE, tras ser enviados y aceptados por la red de medidores.

Para poder enviar los cambios aceptados por la red inteligente y estos queden registrados en la base de datos de CFE se hace uso de un Web Service propiedad de NUBO el cual abre el canal para que Pandora pueda enviar la información correspondiente a los cambios realizados. Cabe resaltar que la información trabajada en las dos bases de datos no es necesariamente una réplica, la base de datos de pandora guarda identificadores y relaciones de los medidores y además de los nuevos dispositivos integrados con el desarrollo de la red inteligente. Por su parte CFE requiere información exclusiva para facturación de servicios, sin importarle información propia de los dispositivos de la red como por ejemplo estatus de conexión y funcionamiento de los Módulos Remotos de Energía o los gabinetes de medidores.

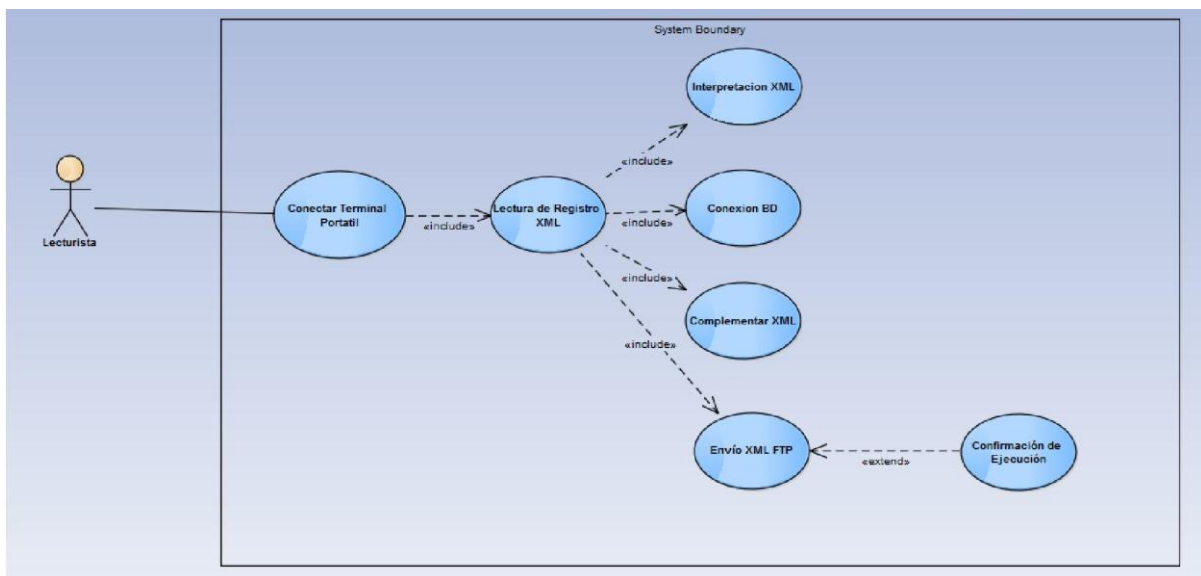


Ilustración 3.10 Diagrama de funcionalidad general para el módulo de terminales portátiles.

Con respecto a la funcionalidad del módulo para terminales portátiles o Hand Held Devices, cabe resaltar que este proceso comienza fuera del Head End con la generación de tareas desde los sistemas informáticos propios de la Comisión Federal de Electricidad, estas son asignadas al Lecturista en un archivo que es enviado a la TP (Terminal Portátil). Así es como el Lecturista inicia su ruta realizando las tareas que le fueron asignadas, registrando los resultados de cada una de estas en una aplicación en la TP.

Una vez que el Lecturista vuelve a la central conecta la terminal portátil a un ordenador vía USB (cuna) para que la información sea descargada. Los datos recabados del registro de actividades del Lecturista es entonces procesado para poder realizar los cambios según corresponda.

Se puede hacer uso de un servicio FTP para enviar los registros hacia pandora y este realice los cambios en su base de datos registrando solamente información pertinente a los dispositivos nuevos integrados a la red de medidores, información útil para estadísticas del mismo sistema o hasta levantamiento de servicios a nuevos usuarios.

El mismo sistema de CFE puede obtener del mismo modo la información referente a la facturación de los servicios o nuevos usuarios integrados para que estos queden registrados en su base de datos

Toda esta interacción será controlada por un Windows Service el cual será descrito a fondo posteriormente en este documento.

El proceso de alta de servicios es de especial importancia para el sistema ya que esta es la razón por la cual se cuenta con una base de datos independiente a la de CFE y NUBO.

El Lecturista desencadena el inicio del proceso al conectar la terminal portátil a la cuna que se encuentra en las oficinas de CFE, el servicio de Windows comienza su ejecución comprobando el correcto estado del documento de registro de actividad generado por el Lecturista, una vez que lo ha comprobado este es procesado generando un documento XML el cual es enviado hacia el Head End para que el modulo del Socket pueda enviar información necesaria para que esta sea guardada en la base de datos de Pandora.

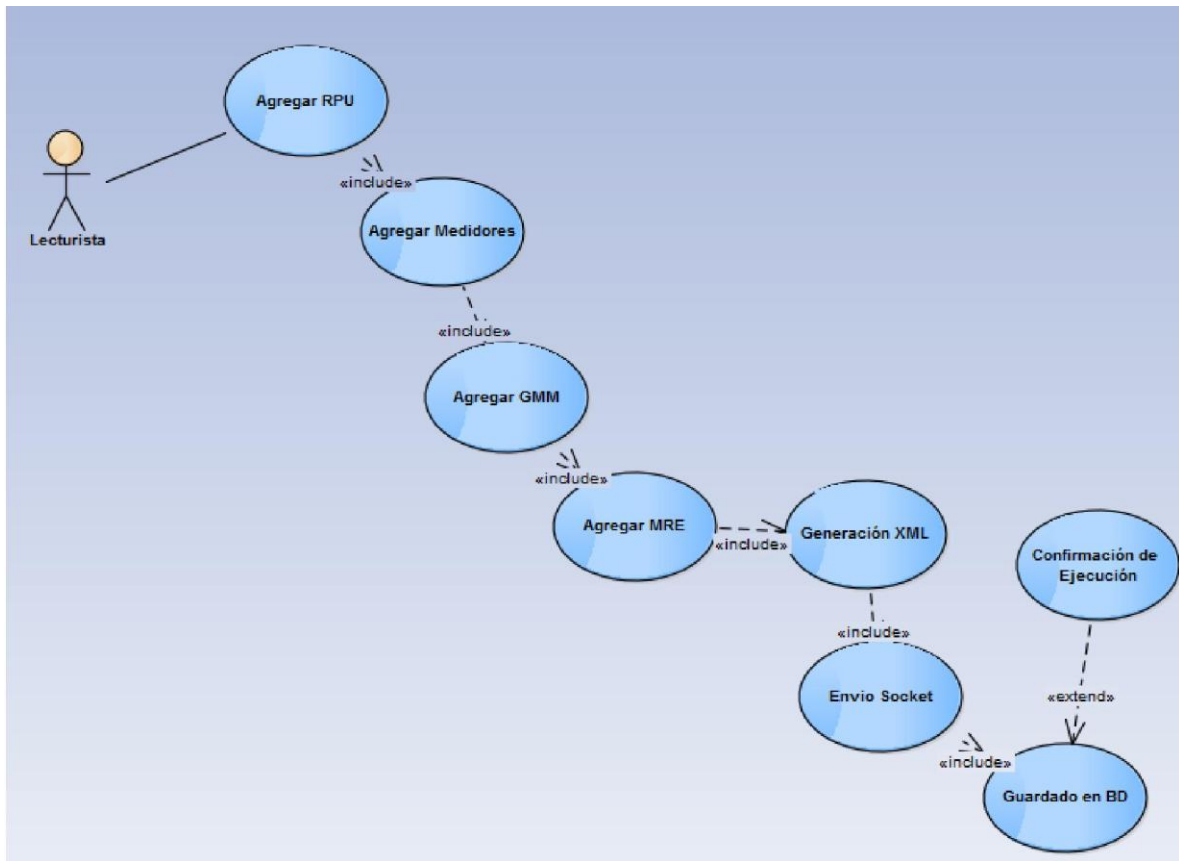


Ilustración 3.11 Proceso para alta de servicio.

De este modo la asociación de los dispositivos que comprenden un servicio:

- Medidores.
- MRE.
- Gabinete.
- RPU.

Generan una asociación única mediante la cual el Head End será capaz de rastrear cada dispositivo de manera individual o a un grupo de dispositivos para poder enviarle un paquete de información. Esto no quiere decir que el Head End guarde una relación de las rutas que deberá de seguir el paquete ya que esta información es almacenada específicamente por el colector de cada segmento de red.

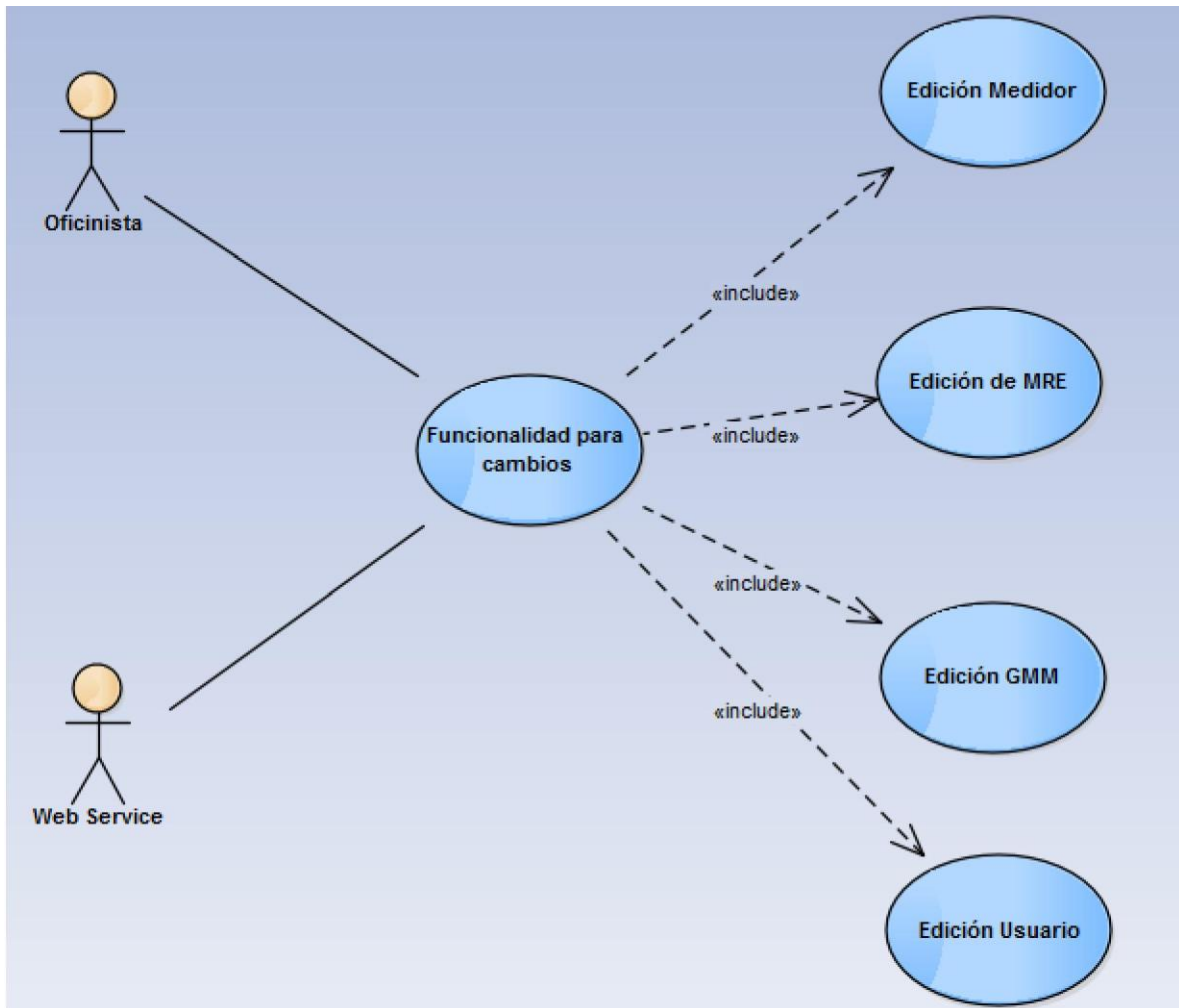


Ilustración 3.12 Sección de cambios en el menú principal del Head End.

Los oficinistas cuentan con una sección dentro del sistema para poder realizar edición en las entidades que conforman la red inteligente.

De este modo se presta una interfaz para que el usuario pueda corregir errores al momento de ingresar los datos identificadores de los dispositivos o de los servicios (RPU), así como de las relaciones que estos tienen.

De la misma manera puede cambiar el estatus de alguno de los dispositivos de manera remota para poder por ejemplo cortar la energía eléctrica a un servicio o a un conjunto de servicios sin necesidad de enviar a un agente de campo para que realice la operación. Los servicios también pueden ser reestablecidos de manera remota en cualquier momento con esta funcionalidad lo que agrega un valor a la prestación del servicio evitando de este modo horas de espera para que la orden fuera levantada y el Lecturista acudiera a realizar la reconexión.

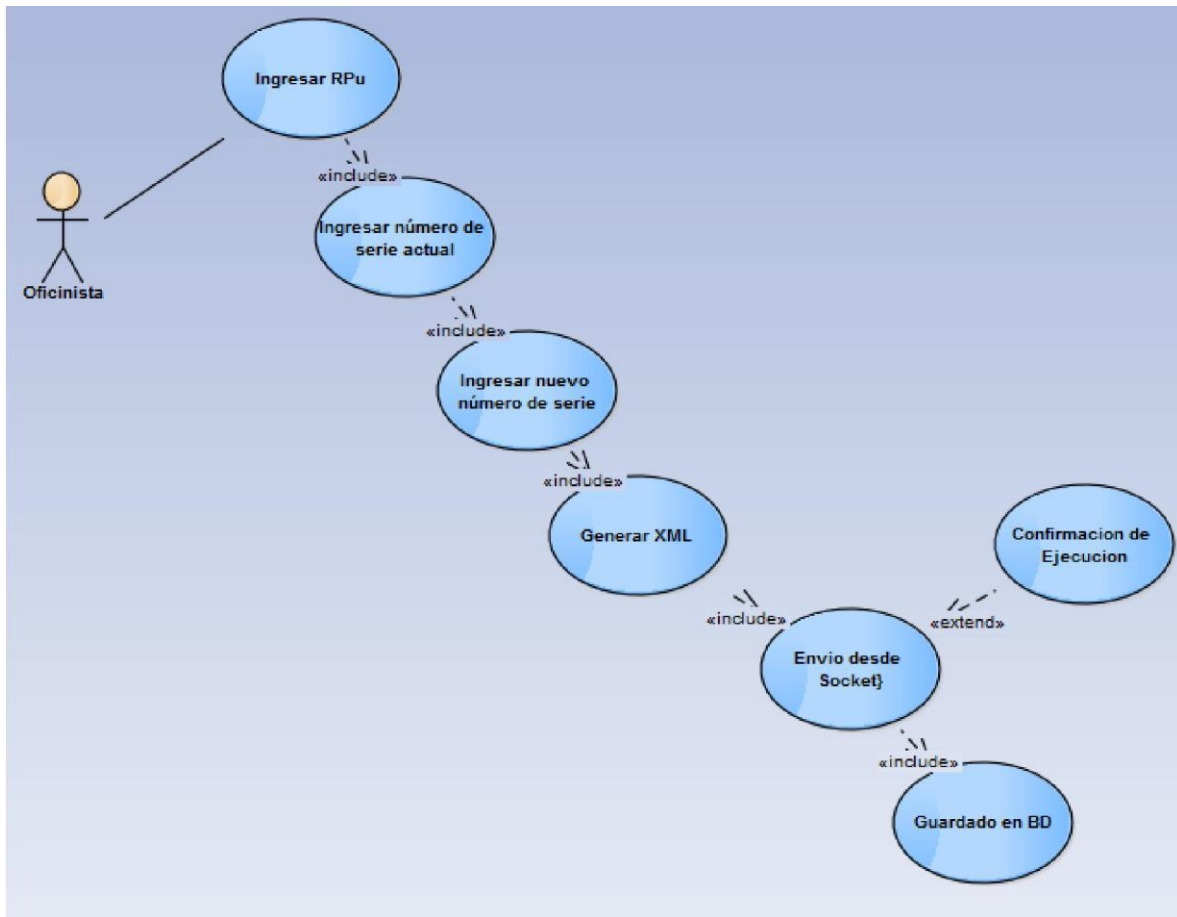


Ilustración 3.13 Diagrama de caso de uso para edición de dispositivo

Para la edición de los datos de alguno de los dispositivos que corresponden a un servicio en específico el oficinista ingresa el primeramente el RPU o identificador del servicio con el cual se desea trabajar, una vez ubicado el dispositivo a editar se ingresa el número de serie actual (autocompletado con una consulta a la base de datos). Al aceptar los cambios se genera un XML el cual es enviado mediante el modulo del Socket hacia la red de medidores para que los cambios sean confirmados se envíe un mensaje de aceptación para que finalmente los registros sean modificados dentro de la base de datos de Pandora.

Cabe resaltar que el proceso de edición de cualquier dispositivo es iniciado por un cambio físico en cualquiera de los dispositivos pertenecientes a un servicio; es decir, la tarea es levantada desde el sistema de CFE para que el Lecturista se dirija hacia la ubicación del servicio a realizar la sustitución, una vez que este vuelve con la confirmación de la operación esta es registrada dentro del sistema y los cambios son registrados en la base de datos.

DIAGRAMAS DE PAQUETES Y CLASES.

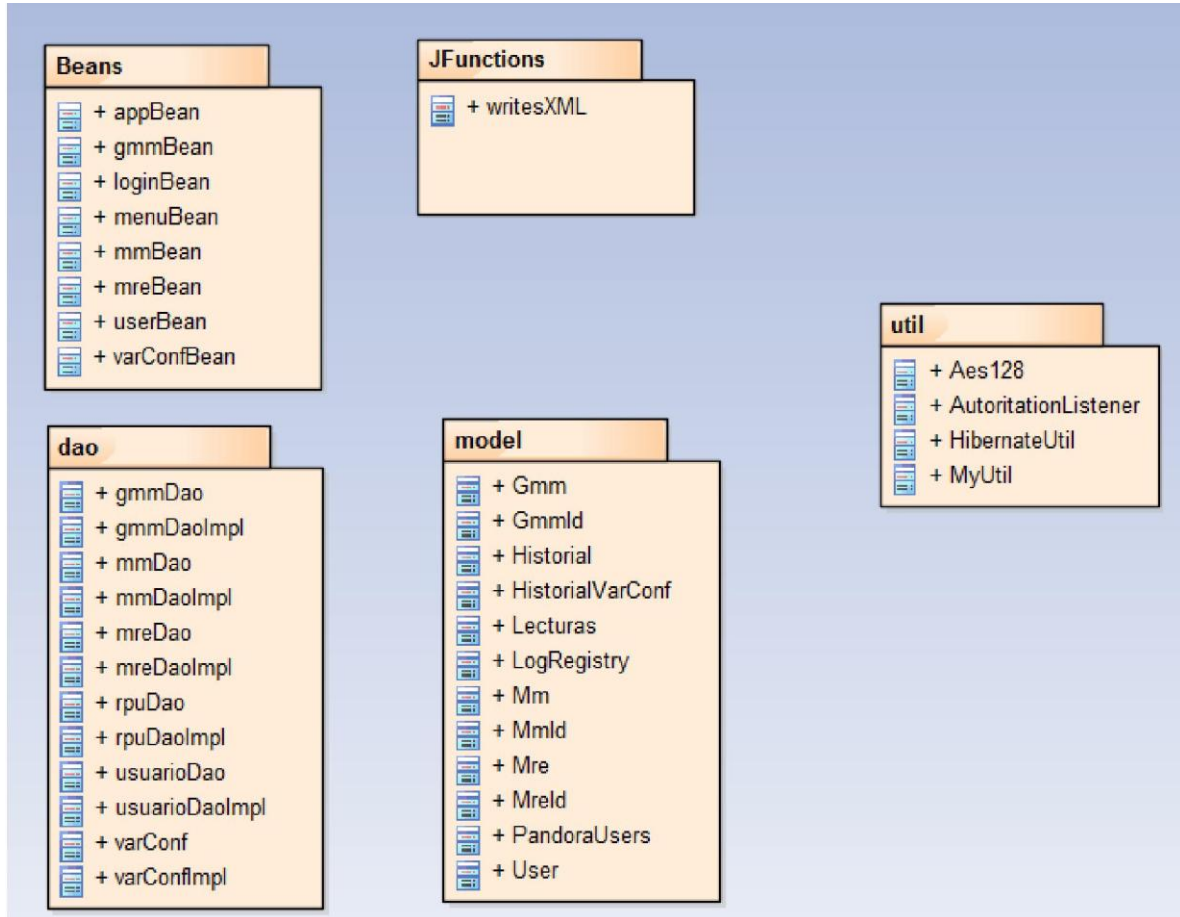


Ilustración 3.14 Diagrama de Paquetes

En la ilustración 3.14 se pueden observar los paquetes que forman parte del Head End Pandora Management System, cada paquete ha sido generado para contener códigos específicos relacionados por su funcionalidad, tales como los modelos de los objetos, las utilidades implementadas dentro del sistema como la funcionalidad para encriptación, clases en Java que contienen funciones complementarias al sistema como la generación de archivos XML para ser enviados a los dispositivos dentro de la red inteligente, y finalmente los Data Objects y Java Beans de los objetos modelados en la base de datos.

Los Java Beans son usados dentro de este lenguaje de programación para la encapsulación de un objeto; en el caso de Pandora los Beans son modelados a partir de las entidades de la base de datos.

Los Beans encuentran sus bases en la programación orientada objetos, por lo cual cada uno de estos cuenta con métodos para establecer (set) y retornar (get) valores referentes a sus propiedades.

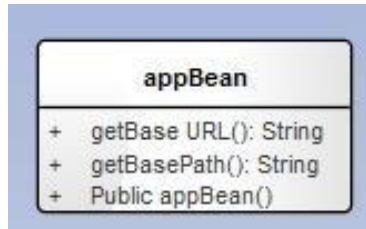


Ilustración 3.15 Diagrama Clase appBean.

En esta clase se puede encontrar los métodos:

- Get Base URL: el cual es una instancia de la clase Java MyUtil (será descrita más adelante en este documento), mediante este método se obtiene una cadena con la URL del sistema.
- Get Base Path: este método retorna en una cadena el Path para la interfaz base del Head End.

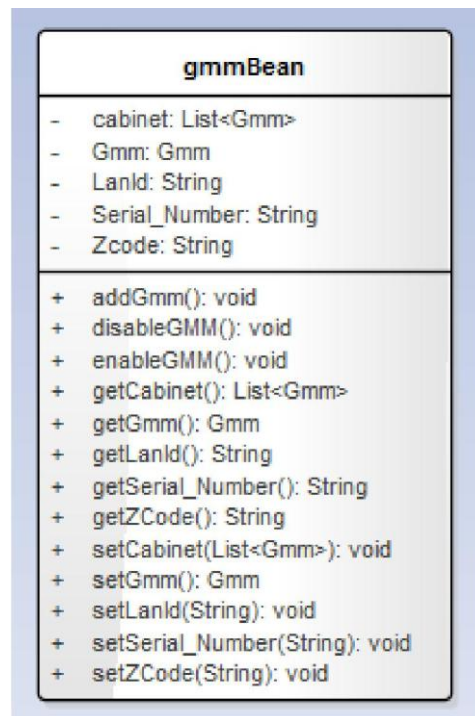


Ilustración 3.16 Diagrama Clase gmmBean.

Las propiedades mencionadas en la parte superior del diagrama de la ilustración 3.16 modelan las columnas creadas para la tabla de gabinetes modulares.

Para cada una de estas propiedades se creó un método para establecer un valor y un método para retornar el valor en base a una consulta a la base de datos.

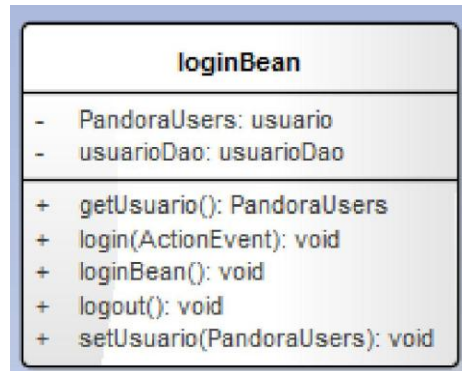


Ilustración 3.17 Diagrama Clase LogInBean.

Está diseñado para realizar una validación de acceso al sistema, una vez que este recibe los datos de la interfaz realiza una consulta a la base de datos para comprobar si el usuario existe dentro de esta y si es que su contraseña coincide con la ingresada por el usuario. En el caso de que el usuario introduzca los datos correctos se le redireccionará a la sección del menú principal del sistema; en caso contrario, se le presentara un mensaje de error para que pueda ingresar sus datos nuevamente.

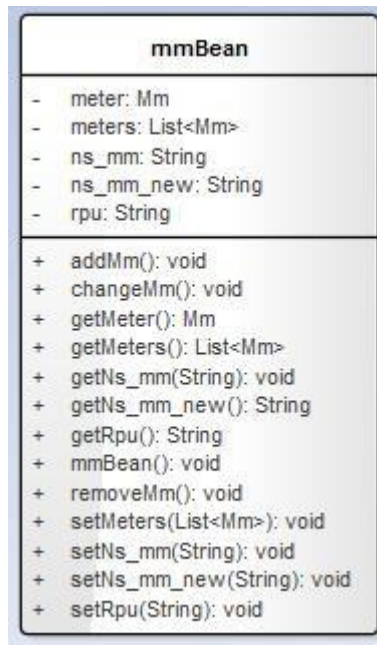


Ilustración 3.18 Diagrama Clase mmBean.

Para controlar las acciones asociadas a los medidores se cuenta con la clase mmBean la cual como puede observarse en la ilustración 3.18 cuenta con métodos para establecer y retornar valores de las propiedades de este objeto así como para realizar una remoción del dispositivo.

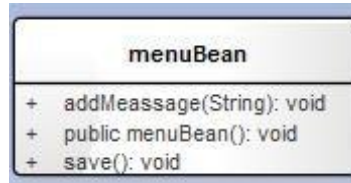


Ilustración 3.19 Diagrama Clase menuBean.

Esta clase es la encargada de presentar mensajes cada vez que una acción es ejecutada, cuenta con mensajes para insertar, actualizar y borrar.

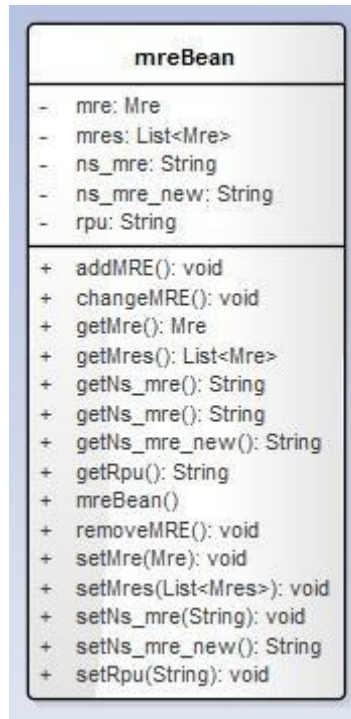


Ilustración 3.20 Diagrama Clase mreBean.

La clase encargada del manejo de los eventos del objeto para los Módulos Remotos de Energía cuenta con métodos para establecer y obtener cada una de las propiedades de este así como un método para la eliminación de este dispositivo en la base de datos.

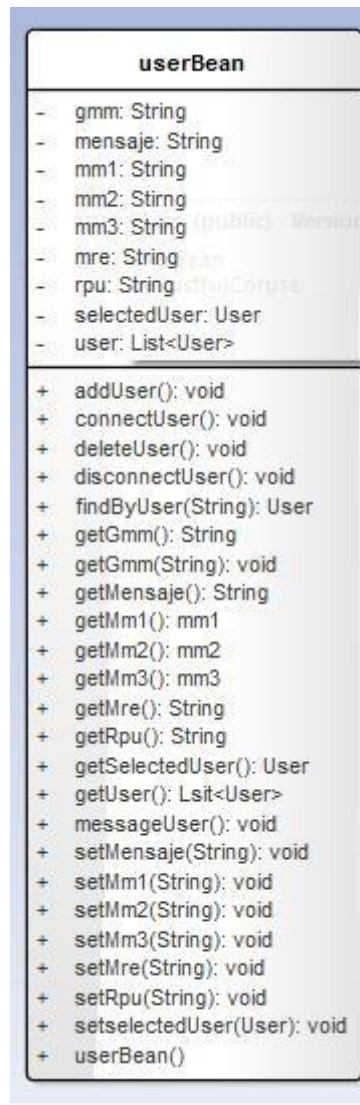


Ilustración 3.21 Diagrama Clase userBean

La clase para el manejo de los usuarios es más compleja en el sentido de las operaciones que deben de ser realizadas para poder consultar o agregar un usuario en específico.

Un servicio está relacionado con diferentes dispositivos dentro de la red inteligente por lo cual habrá que darlos de alta según sean requeridos; por ejemplo, para un servicio monofásico se tendrá un medidor, en caso del bifásico, se cuenta con dos medidores y así sucesivamente. Los demás dispositivos son, el gabinete modular dentro del cual los medidores se encuentran ubicados, un módulo remoto de energía para presentar estimados de consumo.

Esta clase también cuenta con sus propios métodos para establecer y retornar valores a cada una de las propiedades de la entidad además de métodos para enviar mensajes al usuario.

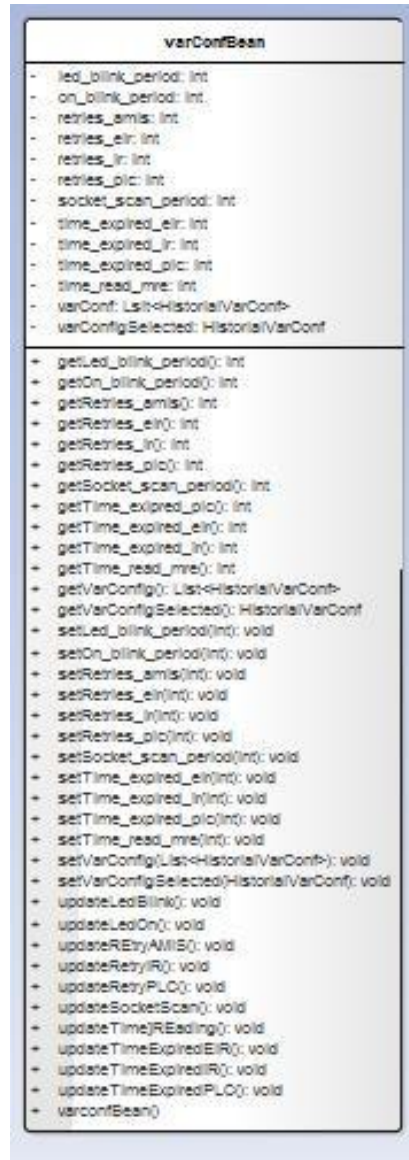


Ilustración 3.22 Diagrama Clase varConfBean.

Las variables de configuración sirven para establecer algunos parámetros de funcionamiento para los dispositivos que integran la red inteligente, por este medio se pueden establecer por ejemplo periodos de lectura, reintentos en envío de mensajes, etc.

La clase `varConfBean` cuenta con métodos para establecer valores a cada una de las propiedades de esta entidad así como métodos para obtenerlos en base a consultas a la base de datos, además de estos, cuenta con funciones para la actualización de los valores que se muestran en la Ilustración 3.22

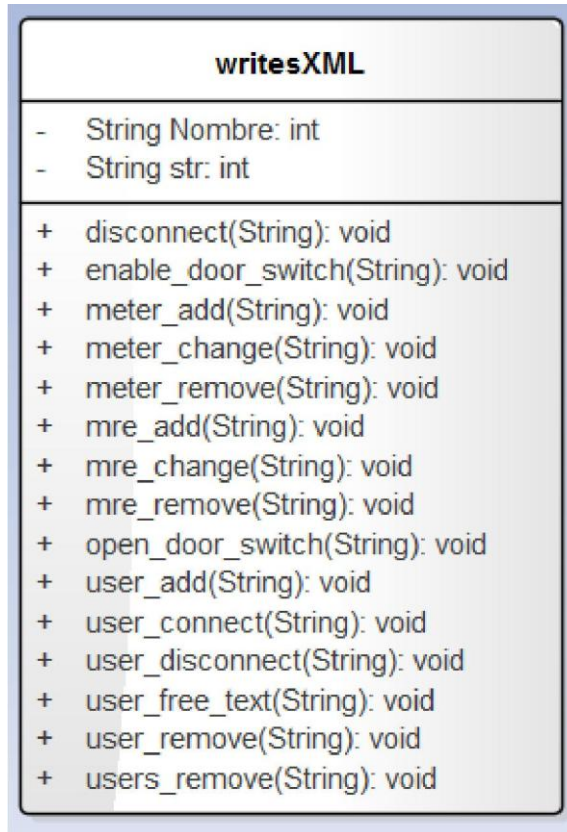


Ilustración 3.23 Diagrama Clase `writesXML`.

Los mensajes que son generados dentro del Head End son enviados en formato XML a los dispositivos dentro de la red de medidores. Esta funcionalidad es complementaria a la del Head End por lo cual se encuentra del paquete `JFunctions`.

Esta funcionalidad del sistema genera un archivo XML con un formato que puede ser interpretado por cada dispositivo dentro del cual se incluye la tarea que se ejecutará por este.

Esta tarea es generada desde la interfaz, así que estos métodos serán invocados desde algún lugar en el módulo web en donde se genera una cadena con los parámetros necesarios para poder crear la tarea y son enviados a este método el cual los concatenará dentro del archivo XML.

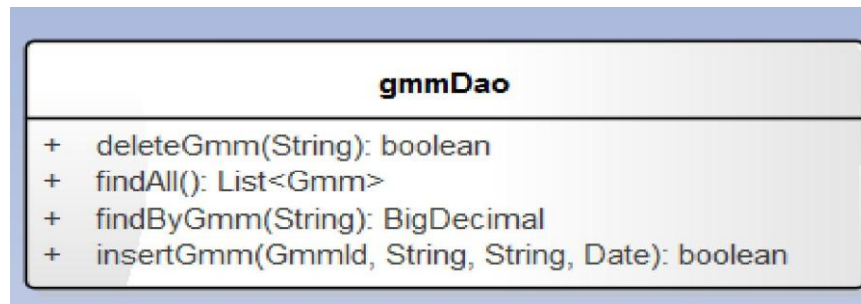


Ilustración 3.24 Diagrama clases gmmDao.

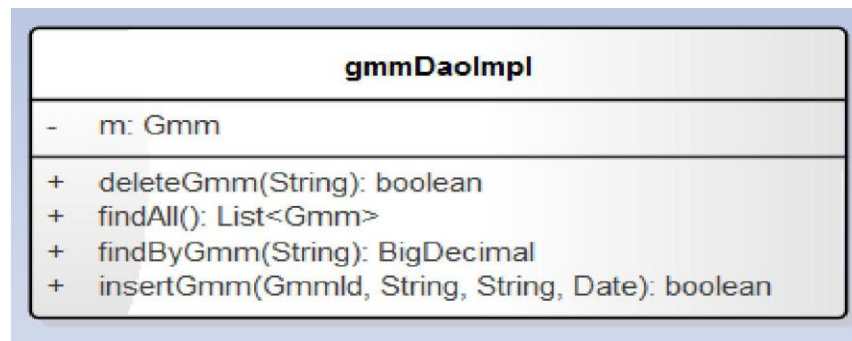


Ilustración 3.25 Diagrama Clases gmmDaoImpl.

Los gabinetes modulares de medidores o GMM son modelados en Data Objects para poder ser manejados dentro de la funcionalidad del sistema.

Para poder realizar operaciones sobre los registros referentes a esta tabla, se han generado dos clases manejadoras una clase llamada fmmDao (Ilustración 3.24) la cual enlista los métodos bajo los cuales se puede operar a estos objetos y la otra gmmDaoImpl (Ilustración 3.25) la cual contiene toda la funcionalidad para que se conecte a la base de datos y realice operaciones de Insertar, Actualizar, Borrar o Seleccionar a la tabla de GMM.

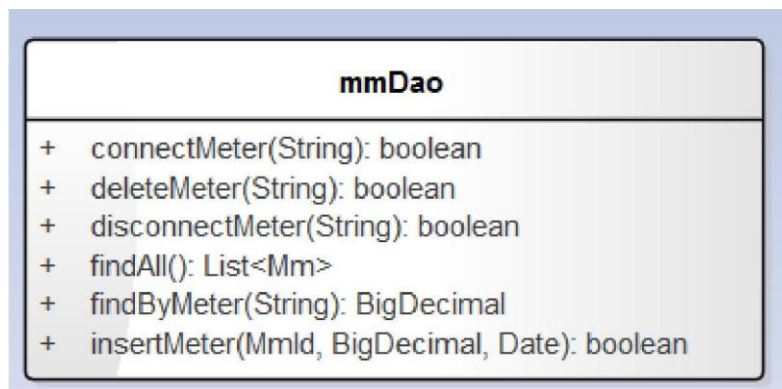


Ilustración 3.26 Diagrama clase mmDao.

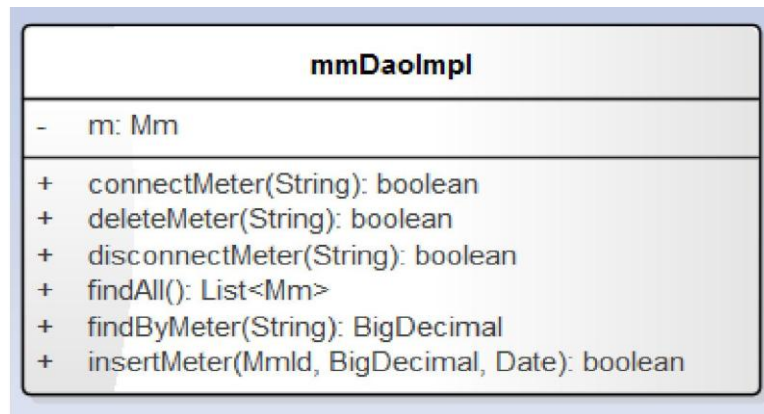


Ilustración 3.27 Diagrama clase gmmDaolmpl.

En la sección de medidores modelado en las clases mmDao y mmDaolmpl se encuentran los métodos creados para la selección de un medidor o una lista de medidores según criterios establecidos por el usuario desde la interfaz gráfica, el método de eliminación de medidor recibe el número de serie del dispositivo que se quiere dar de baja en la base de datos, para la inserción de un medidor se debe de ingresar todos los identificadores de este. Existen dos eventos para la inserción de un medidor, uno es para la sustitución de un dispositivo en caso de malfuncionamiento, o bien la creación de un nuevo servicio lo cual implica la inserción en algunos casos de hasta tres medidores según sea el tipo de servicio.

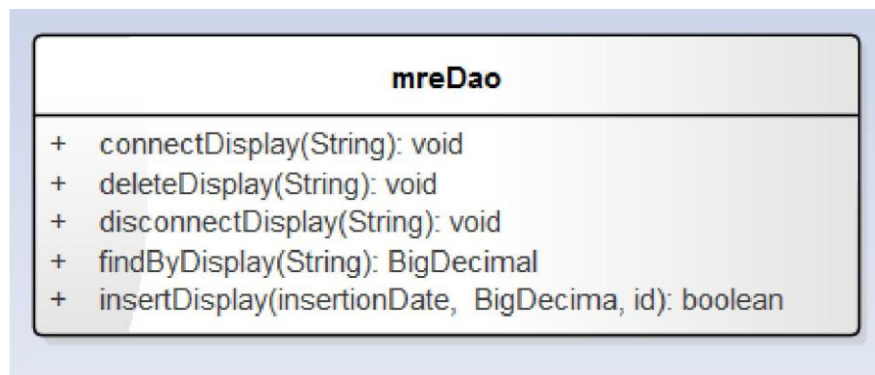


Ilustración 3.28 Diagrama clase mreDao.

Las clases encargadas de las modificaciones a la tabla de Módulos Remotos de Energía mostradas en las Ilustraciones 3.28 y 3.29 contienen métodos para la conexión y desconexión de un MRE de manera remota, es decir enlazar y desenlazar el dispositivo para que éste deje de tener comunicación con los medidores o bien se conecte a los pertenecientes al servicio con el cual se encuentra relacionado.

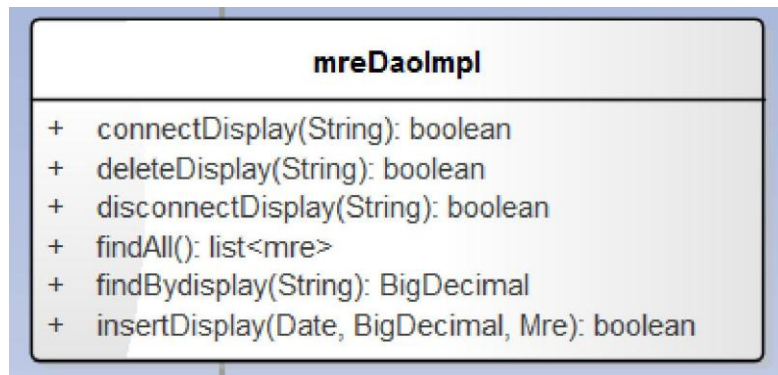


Ilustración 3.29 Diagrama clase mreDaolmpl.

Para consultar MREs desde la interfaz se cuenta con métodos de listado de dispositivos existentes o consulta de información de un registro en particular según sean ingresados los criterios de búsqueda por el usuario.

La clase mostrada en la ilustración 3.29 es la encargada de listar las funciones programadas dentro de la clase mreDaolmpl que mantiene una conexión a la base de datos para retornar la información solicitada desde la interfaz web del Head End.

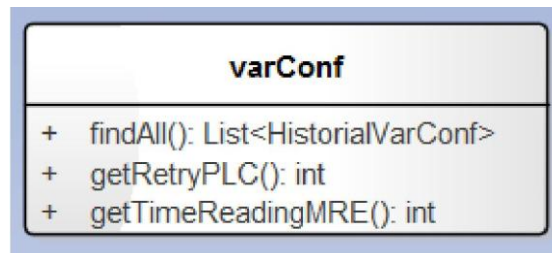


Ilustración 3.30 Diagrama clase varConf

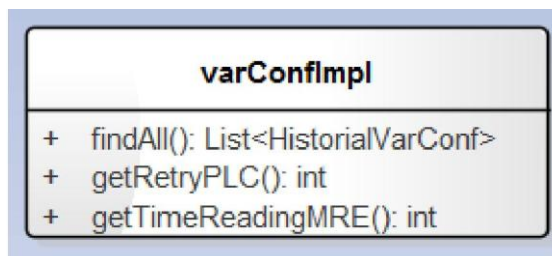


Ilustración 3.31 Diagrama clase varConfImpl

Las clases presentadas en las ilustraciones 3.30 y 3.31 representan el listado de operaciones y la implementación de estas respectivamente. Las variables de configuración como se había mencionado con anterioridad, establecen parámetros de inicio para la funcionalidad de los dispositivos que integran la red inteligente.

Dentro de estas clases tenemos funcionalidad para configurar ciertos parámetros como el número de reintentos para establecer comunicación entre los dispositivos o establecer tiempos de lectura, además de poder obtener un listado de los registros referentes a esta tabla desde la base de datos.

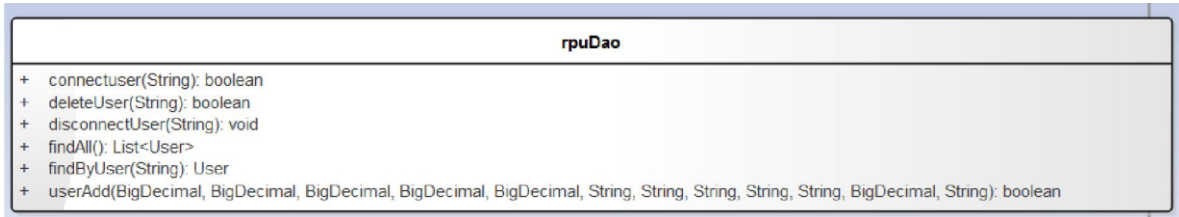


Ilustración 3.32 Diagrama clase rpuDao

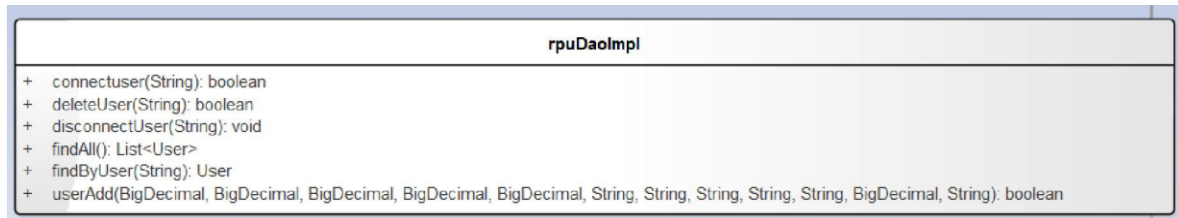


Ilustración 3.33 Diagrama clase rpuDaoImpl.

RpuDao y rpuDaoImpl son las clases encargadas de manejar la funcionalidad para los usuarios de una manera general, lo cual implica una serie de validaciones en los métodos de altas y bajas. Para poder dar de alta un usuario se debe de comprobar si los dispositivos que le están siendo vinculados están dados de alta en sus respectivas tablas, en caso de que estos no existan se deberá de informar al usuario del error ocurrido.

Del mismo modo el método de borrado de usuarios deberá de realizar la eliminación o desconexión de los dispositivos vinculados con el RPU o número de servicio asignado por la Comisión Federal de Electricidad.



Ilustración 3.34 Diagrama de clase userDao.



Ilustración 3.35 Diagrama de clase userDaolmpl.

El sistema requiere de autenticación de usuarios, por lo que se generó la tabla de usuarios, para poder diferenciar entre los usuarios del sistema y los usuarios del servicio de CFE, se denominó a las tablas RPU y PandoraUsers respectivamente. Dentro de las clases mostradas en la ilustración 3.34 y 3.35 se encuentran los métodos para obtener un listado de los usuarios del sistema así como el método de Login el cual se encarga de obtener los datos ingresados por el usuario para enviarlos a la sección de validaciones de acceso descrita anteriormente en este capítulo.

Dentro del paquete “model” podemos encontrar el modelado de los objetos generado desde el Framework Hibernate en este encontramos dos clases por cada entidad de la base de datos.

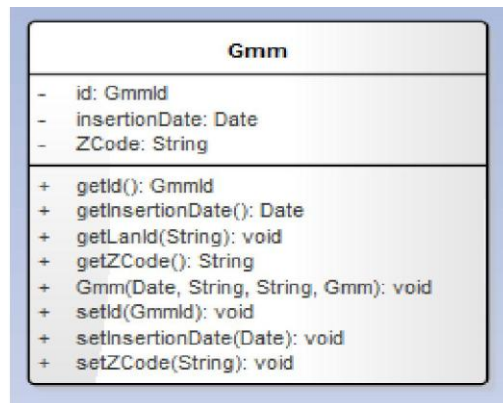


Ilustración 3.36 Diagrama clase Gmm.

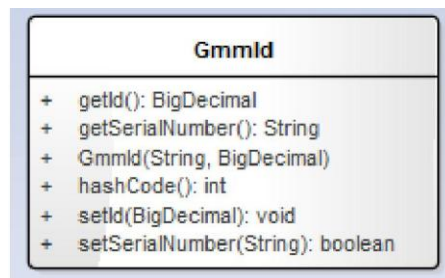


Ilustración 3.37 Diagrama clase GmmId.

Los modelos generados automáticamente para la entidad de Gabinete Modular de Medidores contiene funciones para establecer y obtener valores de las columnas de esta tabla en la base de datos, a diferencia de los Data Objects estos métodos no cuentan con la funcionalidad para retornar listas de elementos.

En la clase Gmmid (Ilustración 3.37) podemos observar que el método getId devuelve un objeto del tipo BigDecimal, el cual contiene las llaves primarias de la entidad modelada.

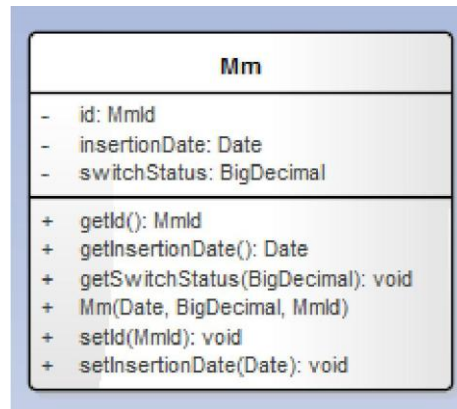


Ilustración 3.38 Diagrama de clase Medidor Modular.

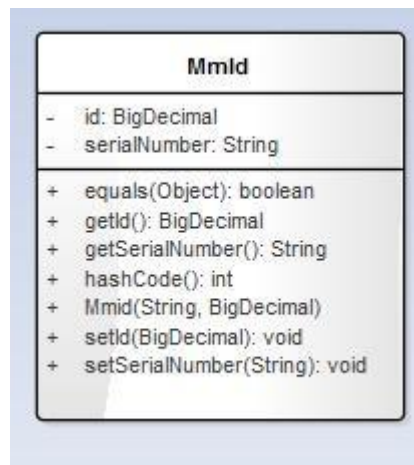


Ilustración 3.39 Diagrama de clase Medidor Modular ID.

Las clases generadas a partir de la entidad que modela a los Medidores Modulares se muestran en las ilustraciones 3.38 y 3.39, donde se pueden observar los métodos para establecer y retornar los valores de sus atributos realizando la instanciación de dichas clases en cualquier parte del sistema, los identificadores únicos de esta entidad se trabajan del mismo modo en un tipo de dato Big Decimal.

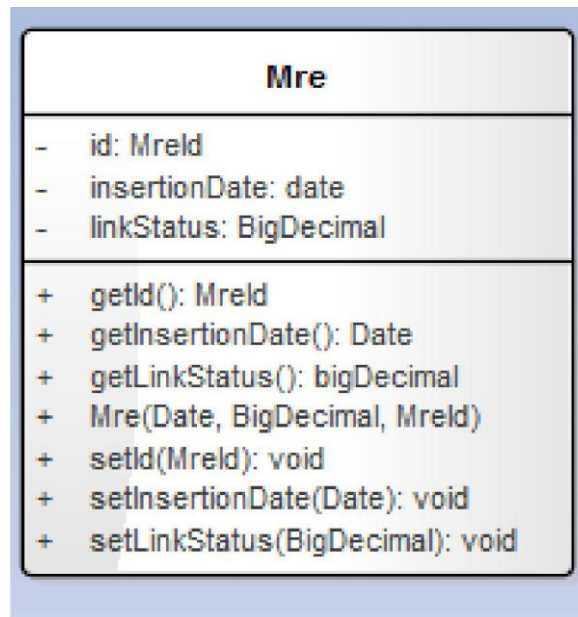


Ilustración 3.40 Diagrama de clase de Módulo Remoto de Energía.

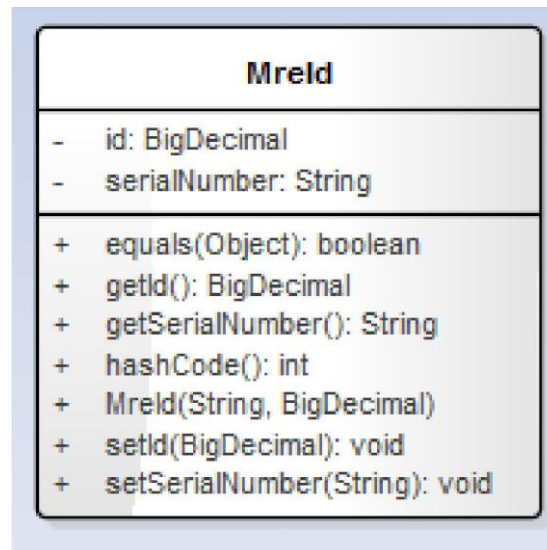


Ilustración 3.41 Diagrama de clase Módulo Remoto de Energía ID.

La entidad Módulo Remoto de Energía fue modelada en Hibernate en dos clases que contienen métodos para poder trabajar sus atributos dentro del Head End, estas clases mostradas en las ilustraciones 3.40 y 3.41 se encuentran dentro del paquete Model que forma parte de la estructura del módulo Web.

Por su parte la entidad que modela las lecturas cuenta con la clase Lecturas mediante la cual se puede acceder a sus atributos de manera individual dentro de la aplicación Web, esta tabla fue modelada en una sola clase a diferencia de las entidades anteriores.

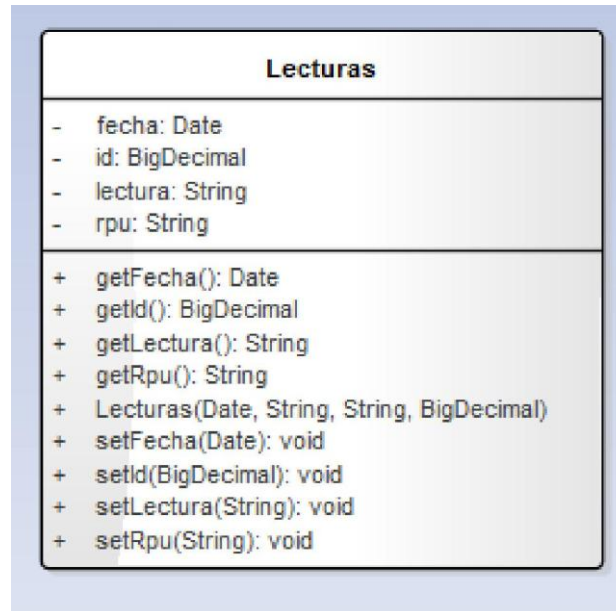


Ilustración 3.42 Diagrama de clase Lecturas.

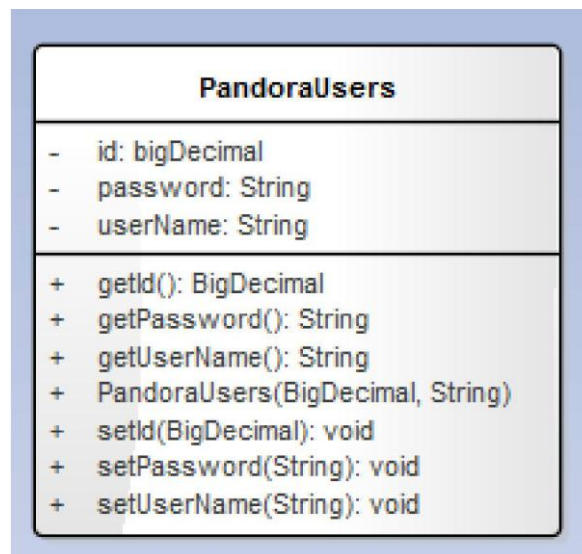


Ilustración 3.43 Diagrama de clase Pandora Users.

Para el manejo de la entidad de los usuarios del sistema existe la clase Pandora Users mostrada en la ilustración 3.43, la cual contiene métodos para asociar valores

ingresados por el usuario en el sistema a los atributos del objeto modelado, al igual que métodos para consultarlos.

La tabla encargada de manejar los registros referentes a los cambios en las variables de configuración de los dispositivos fue modelada en una clase la cual es mostrada en la ilustración 3.44, esta clase es la responsable de brindar acceso al usuario a los atributos de la tabla en la interfaz web del Head End.

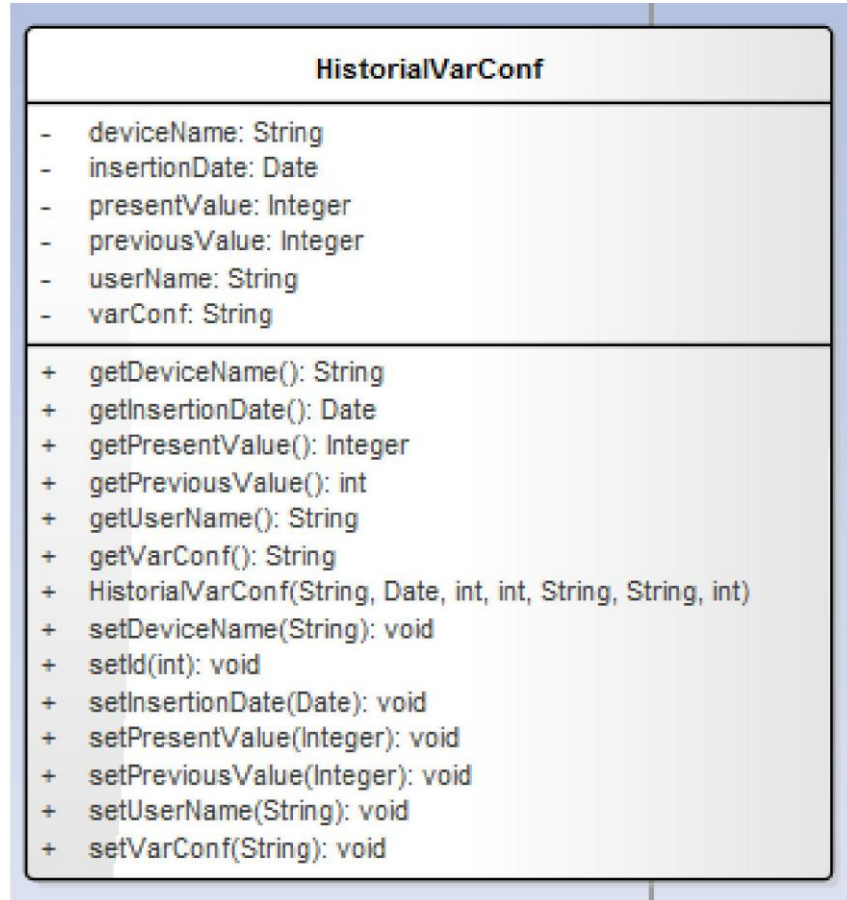


Ilustración 3.44 Diagrama de clase Historial de Variables de Configuración.

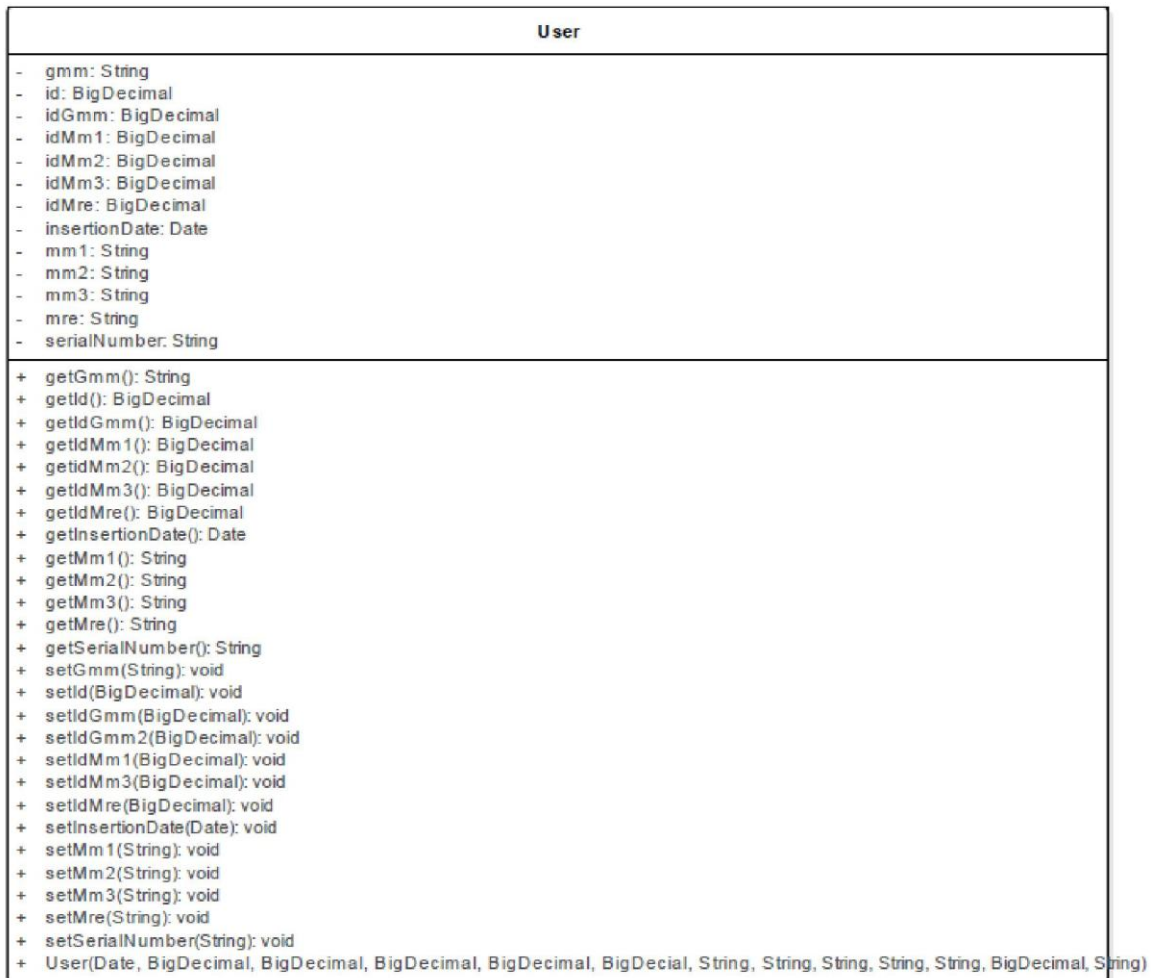


Ilustración 3.45 Diagrama de clase User.

Finalmente la entidad usuario, previamente mencionada como RPU en la sección de Objetos de datos, fue modelada por el Framework en una clase la cual cuenta con métodos para poder acceder a los atributos de la tabla enlistados en la ilustración 3.45. Al igual que en las clases contenidas en el paquete DAO, se cuenta con métodos para cada uno de los dispositivos asociados al servicio, lo cual facilita el acceso a todos los objetos relacionados con esta tabla.

DIAGRAMAS DE ESTADOS.

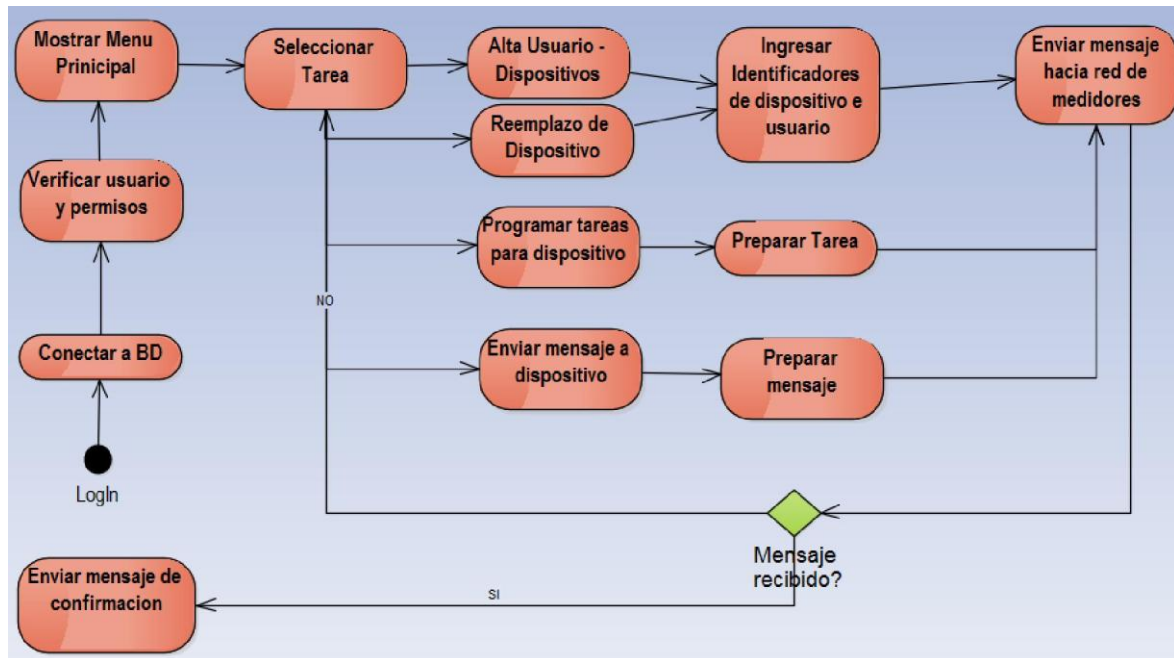


Ilustración 3.46 Diagrama de estados de la funcionalidad generalizada para el módulo de interfaz Web.

El módulo web encargado de brindar una interfaz para el usuario de oficina de la comisión federal de electricidad sigue básicamente la serie de estados mostrada en la ilustración 3.46 para describir de manera generalizada su funcionamiento.

El proceso comienza con el acceso al sistema mediante la ventana de Log In la cual realiza una conexión a la base de datos para validar la entrada del usuario al sistema, una vez que los datos son traídos desde la BD, se verifican sus permisos para poder mostrar el menú principal en donde se encuentran todas las funcionalidades mencionadas en secciones anteriores de este capítulo.

Dentro del menú principal el usuario puede seleccionar la tarea a realizar que a manera de resumen puede ser:

- Alta de usuario o dispositivo.
- Reemplazo de dispositivo.
- Programar tareas (variables de configuración).
- Enviar mensaje a dispositivo.

Para dar de alta un usuario o dispositivo el oficinista deberá de ingresar los atributos identificadores del dispositivo los cuales son mostrados en un formulario.

La tarea de reemplazo de dispositivo es semejante a la de altas, con la diferencia de que el usuario deberá de seleccionar un RPU, con lo cual podrá observar los dispositivos asociados a ese servicio y de este modo seleccionar uno para realizar el reemplazo de este agregando en los campos requeridos la información necesaria de identificadores del dispositivo.

La generación de tareas se inicia cuando el oficinista selecciona un servicio ingresando el RPU, una vez identificado el servicio se procede a ingresar los parámetros deseados para la configuración de este.

El proceso de envío de cualquier mensaje hacia la red de medidores implica la intervención del módulo del Socket TCP el cual es el encargado de direccionar los paquetes hacia el colector para que este pueda enviarlos hacia el o los dispositivos correctos. Los mensajes cuentan con un formato específico listo para ser interpretado por cada uno de los dispositivos que integran la red de medidores inteligente.

Una vez que el mensaje ha sido recibido por el dispositivo al que fue enviado se genera mensaje con una respuesta de confirmación lo cual permite que los cambios en la base de datos sean realizados manteniendo un registro de cambios efectuados.

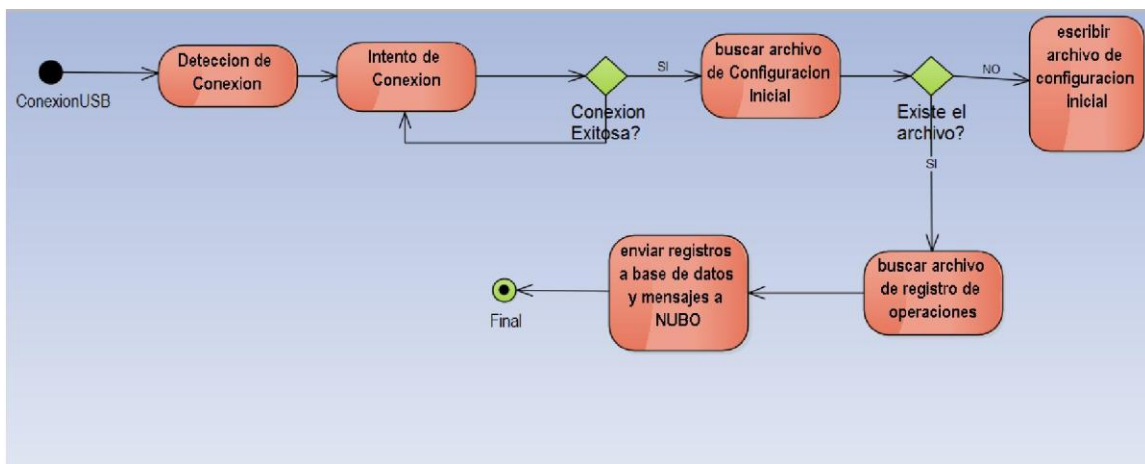


Ilustración 3.47 Diagrama de estados para la funcionalidad generalizada del módulo de conexión con terminales portátiles.

EL servicio encargado de detectar la conexión de las terminales portátiles para la obtención del registro de tareas desempeñadas en campo sigue una serie de estados presentados en la ilustración 3.47.

Este servicio o "Daemon" (Demonio) comprueba de manera periódica la conexión de dispositivos en los puertos USB, la primer tarea ejecutada es buscar el archivo de configuración de la TP el cual no existirá si se trata de la primera vez que es insertado el dispositivo; si esto es así, el servicio crea y envía el archivo a los directorios de la Hand Held.

Una vez que se ha realizado esa comprobación se prosigue con la recuperación del archivo que contiene el registro de operaciones que el Lecturista realizó durante su jornada de trabajo.

Este archivo será procesado por el mismo demonio para ser enviado según corresponda hacia NUBO para que envíe la información para ser registrada en la base de datos de CFE o bien al Head End Pandora para que este realice los cambios necesarios en su propia BD.

HEAD END – PANDORA MANAGEMENT SYSTEM.

Un Head End es el sistema informático encargado de la vinculación de los dispositivos que integran una red inteligente de medidores con los sistemas propios de la Comisión Federal de Electricidad.

Este sistema debe de proporcionar a CFE interfaces para que sus operadores puedan interactuar de manera remota con los dispositivos y registrar los resultados de tareas de campo, por lo cual la descripción de su desarrollo se tratará acorde a los módulos que se programaron para cubrir estas necesidades.

Un módulo web fue desarrollado para cubrir la necesidad de una interfaz por la cual el trabajador de oficina de la Comisión Federal de Electricidad pueda generar tareas, enviar información de configuración o peticiones a los dispositivos dentro de la red de medidores, además de poder consultar datos estadísticos sobre el uso del sistema mismo y del funcionamiento de la red inteligente.

El proyecto de desarrollo de Head End, debido a sus dimensiones fue dividido en dos fases. La primera fase es el desarrollo de la funcionalidad básica para levantamiento de tareas, comunicación con la red de medidores y comunicación con terminales portátiles.

En la segunda fase, se desarrollará funcionalidad complementaria para consultas estadísticas sobre el funcionamiento del sistema y de la red de medidores; esta información será analizada a mayor profundidad más adelante en este documento en la sección de trabajos futuros.

MÓDULO WEB PANDORA WEB GUI.

La interfaz web se inició como un Servlet el cual trabajaría instalado en un servidor remoto al cual ingresarían los oficinistas de la comisión, con el progreso del desarrollo y en miras a una mejor adaptación a los requerimientos presentados por el propietario del producto se migró la funcionalidad a tecnologías JSP; por estas razones, el lenguaje de programación seleccionado para este desarrollo fue Java.

Este módulo es accesado desde una URL desde cualquier central de CFE y soporta accesos simultáneos. La primer tarea desempeñada es brindar un acceso validado mediante un apantalla de Log In, dentro de la cual el usuario ingresara sus datos de usuario y contraseña para poder tener acceso al menú principal del sistema.

Una vez dentro del sistema, en el menú principal se presenta la funcionalidad esencial del Head End la cual comprende:

- Dispositivos.
- Usuarios.
- Facturación.
- Alarmas.
- Variables de Configuración.
- Reportes.

En el Menú de Dispositivos se encuentra un submenú para cada uno de los dispositivos que integran la red inteligente de medidores con funcionalidad para agregar, cambiar y remover.

En la sección de usuarios se encuentran los siguientes submenús:

- Nuevo.
- Servicio. Que contiene funcionalidad para conectar y desconectar.
- Mensajes.
- Eliminar. Con funcionalidad para eliminar ya sea uno o varios al mismo tiempo.

En facturación los submenús comprenden las siguientes tareas:

- Asignar datos RPU
- Asignar tarifa RPU.
- Remover datos RPU.
- Remover tarifa RPU.
- Remover tarifas.

El menú de Alarmas contiene funcionalidad para:

- Puerta abierta.
- MMs No autorizados.
- MMs huérfanos.
- Falla M13.
- Error memoria EEPROM.
- Temperaturas.
- Error con MM.
- Nivel de señal PLC.

En cuanto a las variables de configuración, las tareas programadas comprenden:

- CCG.
 - a. Periodo de lectura MRE.
 - b. Reintentos PLC.
 - c. Tiempo expirado PLC.
 - d. Reintentos IR.
 - e. Tiempo expirado IR.
 - f. Reintentos EIR.
 - g. Tiempo expirado EIR.
 - h. Periodo escaneo de conector.
 - i. Periodo de parpadeo del LED.
 - j. Periodo de parpadeo del encendido.
 - k. Reintentos AIMS.
- MRE
 - a. Tiempo habilitado botón.
 - b. Tiempo deshabilitado botón.
 - c. Tiempo muestra símbolo.
 - d. Tiempo cambio de lectura.
 - e. Tiempo cambio de lectura manual.

- f. Tiempo parpadeo símbolo PLC.
- g. Intentos para configurar AMIS.
- h. Tiempo expiración del CCG.

El menú de reportes contiene las siguientes tareas:

- Nuevos servicios.
- Desconexiones.
- Conexiones.
- Alarmas.
- MMs No autorizados.
- MMs huérfanos.
- Falla en MM13.
- Temperaturas.
- Error con MM
- Nivel de señal de PLC.
- Error de memoria EEPROM.
- MMs ABC.
- MREs ABC.
- Mensajes RME.
- Log de Operaciones.

3.12 HEAD END DAEMON.

Para el módulo de conexión con las terminales portátiles se desarrolló unWindows Service en el lenguaje de programación C# con conexión a base de datos de Oracle y uso de archivos XML

El servicio se ejecuta en segundo plano dentro del sistema operativo del computador donde reside. Brindará la ventaja en cuanto a que su funcionamiento es completamente transparente para el usuario, lo cual evita que este realice la misma tarea de manera rutinaria cada vez que un dispositivo sea conectado al ordenador; facilitando de esta manera su trabajo y ahorrándole tiempo.

El proceso para este software iniciara cuando SICOM genera una orden de trabajo que contiene rutas y tareas para el personal de campo y es descargada a un Hand Held Device para que pueda ser portátil.

Recibida la orden de trabajo el personal se dirigirá a campo a realizar sus actividades asignadas guardando un registro con los resultados de estas.



TECNOLÓGICO NACIONAL DE MÉXICO
Instituto Tecnológico de Apizaco

"2015. Año del Generalísimo José-María Moreles y Pavón"



De vuelta en la central conectará su Hand Held Device con el resultado de las tareas a un computador designado donde se encontrará residiendo este software.

Este software deberá de ser capaz de detectar que el dispositivo ha sido conectado, explorar dentro de sus directorios hasta hallar el fichero CSV o delimitado por comas, el cual contiene los resultados de las tareas, interpretarlo y registrar dichos resultados en una base de datos.

Esta base de datos es la misma que usa el software PANDORA, se hace uso de una clase encargada de abrir, cerrar la conexión así como ejecutar Queries para hacer SELECT, DELETE, y UPSERT que como se ha mencionado con anterioridad es una operación encargada de comprobar si el registro existe y en su caso actualizar o insertar uno nuevo.

Tras haber guardado los registros en la base de datos, debe de generar un archivo XML para que NUBO tenga conocimiento del resultado de las operaciones y tenga también un registro de lo hecho durante las transacciones.



TECNOLÓGICO NACIONAL DE MÉXICO
Instituto Tecnológico de Apizaco

"2015. Año del Generalísimo José-María Moreles y Pavón"



CAPITULO IV PRUEBAS Y RESULTADOS

4.1 RESULTADOS

En el presente capítulo se muestran imágenes de la interfaz del Head End Pandora Management System en un entorno de pruebas, resultado del desarrollo alcanzado por el equipo de programación; así como diagramas de Gantt que muestran el avance en relación a fechas durante el periodo de duración de las estancias.

El Sistema es ejecutado desde un servidor con 8gb de memoria RAM capaz de ejecutar el gestor de bases de datos al mismo tiempo que el Head End sin ningún problema. Para esta fase, se hace uso de información aleatoria para trabajar los registros con los cuales funciona el sistema; esta información se tomó de medidores y gabinetes de pruebas dentro de Softek Global eDesign elegidos para ser usados dentro de la misma empresa simulando un entorno de funcionamiento real, lo cual presenta información con dimensiones y caracteres reales para la ejecución de simulaciones lo más cercanas a la realidad.

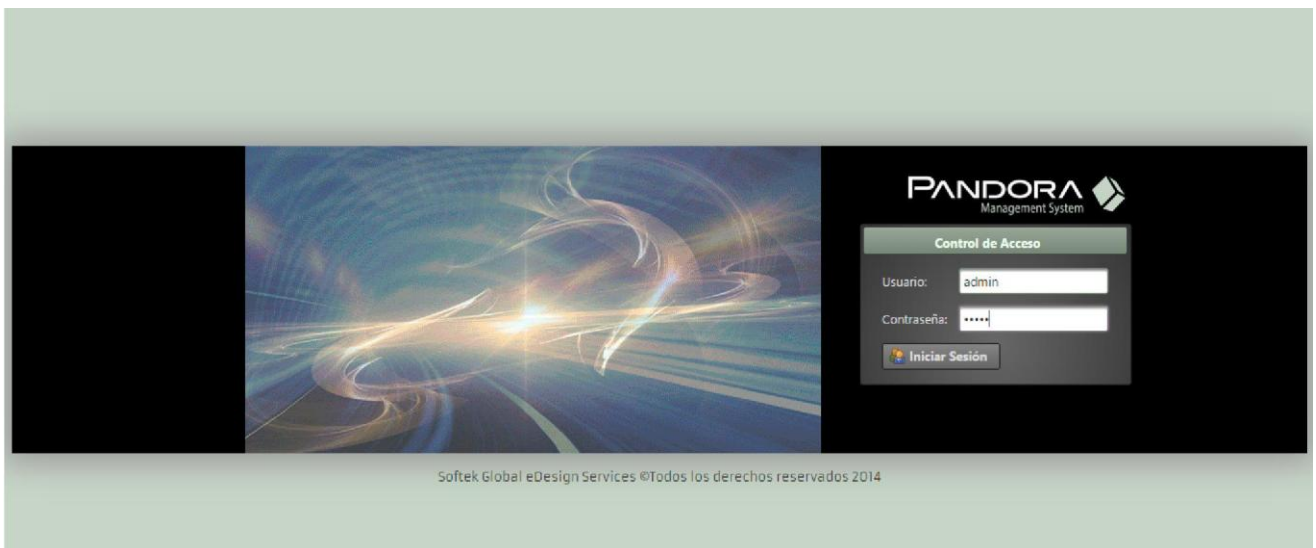


Ilustración 4.1 Captura de pantalla de LogIn.

El usuario accede mediante una URL (Uniform Resource Locator) de manera remota a una instancia del sistema el cual es ejecutada desde el servidor donde este se encuentra instalado, presentando de inicio una interfaz mediante la cual el usuario puede ingresar sus datos identificadores como usuario y contraseña para poder validar su acceso al sistema (ilustración 4.1), lo cual brinda un primer parámetro de protección para el sistema al ofrecer acceso solo a usuarios registrados dentro de la base de datos con un estatus de activo.

En caso de que los datos ingresados no sean validados de manera correcta, el usuario no podrá acceder al menú principal de la aplicación.

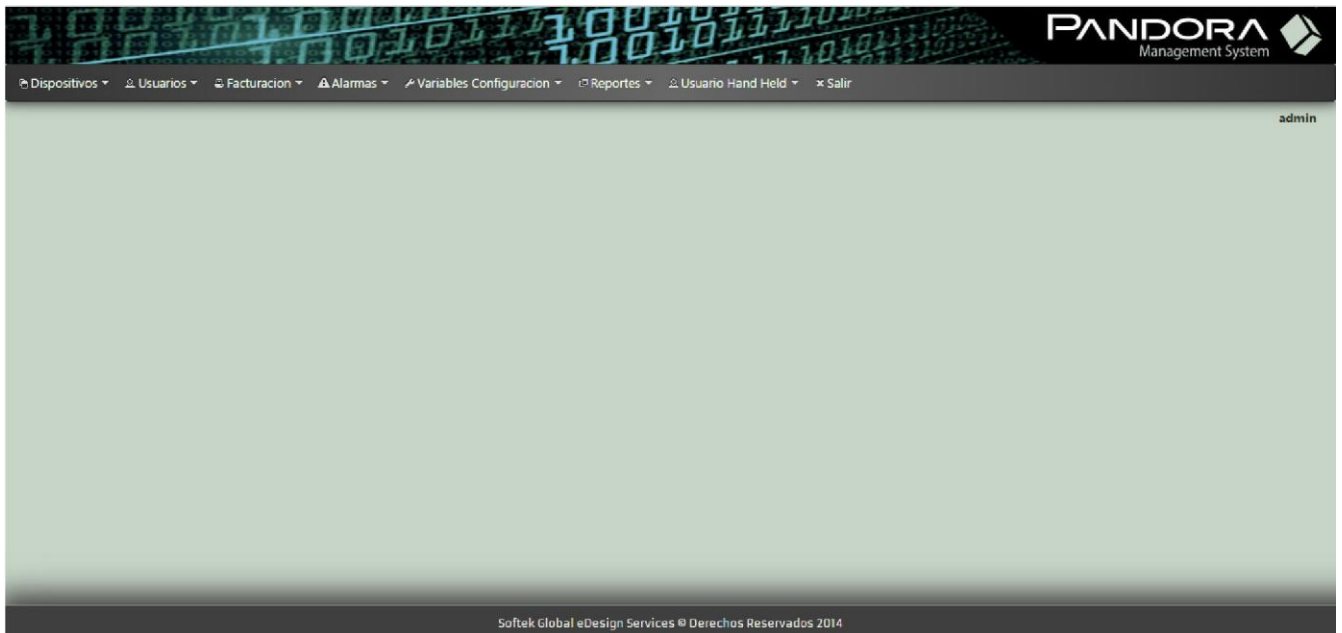


Ilustración 4.2 Menú Principal del sistema.

Una vez que el usuario ha ingresado sus datos de manera correcta, la interfaz en la ilustración 4.2 que le es presentada es la del menú principal del sistema, donde se puede navegar por toda la funcionalidad que el sistema le brinda, pudiendo de este modo generar las tareas, configuraciones y asociaciones para los dispositivos que integran la red de medidores inteligente. En este menú principal del sistema se presenta en forma de barra de menús un grupo de apartados que distinguen las diferentes entidades con las que puede interactuar así como diversas funcionalidades referentes a registros de uso de los dispositivos e del sistema mismo.

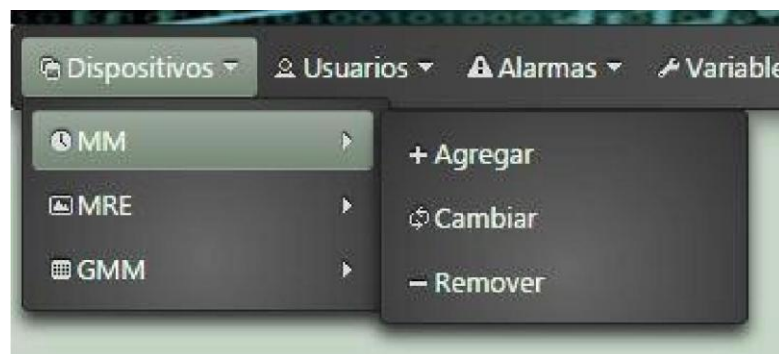


Ilustración 4.3 Tareas para medidores modulares.

El primer apartado al cual se puede ingresar desde la barra de menús de Pandora Management System es la de “Dispositivos”. Esta sección hace referencia a las tareas que pueden ser realizadas sobre cada una de las entidades que forman parte de la red de medidores inteligente, entre ellas tenemos a los medidores modulares o MM, al módulo remoto de energía o MRE y finalmente al gabinete modular de medidores o GMM. El primer submenú corresponde a los medidores modulares los cuales tienen opciones para agregar, cambiar o remover como puede observarse en la ilustración 4.3.



Ilustración 4.4 Interfaz para agregar un medidor.

El procedimiento para dar de alta un nuevo medidor modular dentro del Head End se ejecuta ingresando en la interfaz del sistema dentro de los dos campos de texto presentados, los identificadores propios del dispositivo como se muestra en la ilustración 4.4 y se presiona el botón agregar para enviar los registros a la base de datos.

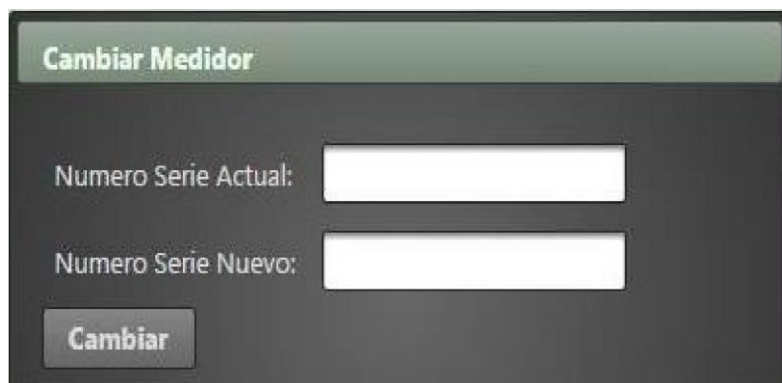


Ilustración 4.5 Interfaz para edición de información de los medidores.

La ilustración 4.5 presenta la interfaz para ingresar los identificadores de un medidor al ser sustituido en un servicio específico, por lo cual el registro del número de serie anterior es enviado a un historial de cambios y de este modo se cambia el nuevo valor en la base de datos.

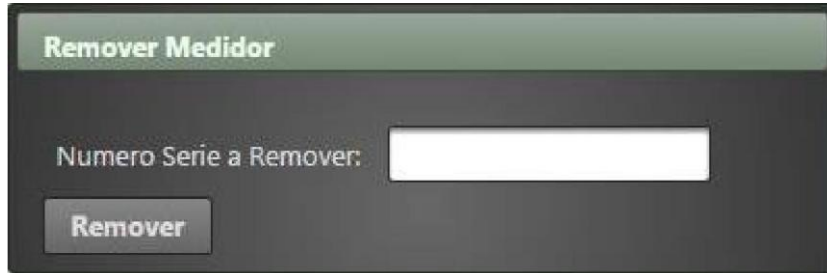


Ilustración 4.6 Ingreso de datos para la remoción de un medidor.

Dentro de las funcionalidades que presenta el Head End se encuentra la de remover un medidor (ilustración 4.6), proceso para el cual se debe ingresar el número de serie del medidor que se desea dar de baja del sistema enviando el valor del identificador del dispositivo a el registro histórico de cambios dándolo de baja de la relación de medidores activos dentro de la red.



Ilustración 4.7 Lista de tareas para dispositivos de tipo MRE.

Los dispositivos de modulo remoto de energía cuentan al igual que los medidores con opciones para la inserción, modificación, y borrado de los registros correspondientes a esta entidad en la base de datos. Como se observa en la ilustración 4.7 se presenta al usuario de oficina una lista desplegable de las operaciones a ejecutar para los MRE.



Ilustración 4.8 Ingreso de datos para agregar un MRE.

El proceso de adición de un nuevo MRE se ejecuta al ingresar los números identificadores del servicio al que se quiere agregar el dispositivo, así como el número de serie único del Módulo remoto de energía lo cual genera un nuevo registro en la tabla correspondiente tras haber enviado los cambios a la red de medidores (ilustración 4.8).



Ilustración 4.9 Ingresar para ingreso de datos de edición de MRE.

En el evento de que un MRE presente problemas de funcionamiento el operador de oficina puede realizar su sustitución ingresando los identificadores del dispositivo accedendo a la sección de cambiar MRE dentro del Head End en donde podrá ejecutar la operación guardando los cambios realizados al presionar el botón "cambiar" como se muestra en la ilustración 4.9.

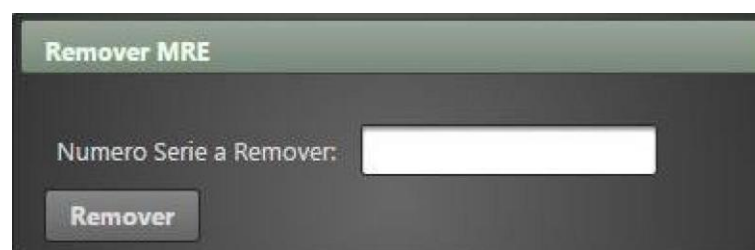


Ilustración 4.10 Interfaz para ingreso de datos para la remoción de MRE.

Una más de las secciones mostrada en la ilustración 4.10, presente dentro del sistema para interactuar con este tipo de dispositivo es la de remover MRE la cual permite al oficinista dar

de baja un dispositivo ingresando solamente el número de serie de este para que el registro deje de ser presentado en la lista de dispositivos activos dentro de las tablas de cada dispositivo.



Ilustración 4.11 Lista de tareas para Gabinetes.

Cuando el usuario ingresa al submenú de gabinetes modulares (ilustración 4.11), se le muestran apartados para realizar tareas de insertar un nuevo elemento al cual serán relacionados los dispositivos mencionados previamente (MM, MRE), y además una sección para configurar alarmas relacionadas directamente con el interruptor de la puerta del GMM.

Ilustración 4.12 Interfaz para agregar un gabinete.

La sección de tareas para los gabinetes modulares comienza con la opción de agregar un nuevo gabinete (ilustración 4.12), la cual presenta una interfaz en la cual el oficinista deberá de ingresar en los campos correspondientes los identificadores de este tipo de dispositivo para poder darlo de alta en la tabla de gabinetes.



Ilustración 4.13 Lista de tareas para el interruptor de la puerta de un gabinete.

Las tareas que se pueden ejecutar con los gabinetes modulares incluyen el cambio de estatus para el interruptor de la puerta (Ilustración 4.13) lo cual dará acceso al lectorista al interior GMM para poder manipular físicamente los medidores dentro de él.

Ilustración 4.14 Ingreso de datos para habilitar el interruptor de puerta.

El submenú habilitar puerta perteneciente a la sección de gabinetes muestra un control de tipo TextBox en donde se debe de ingresar el identificador de red del dispositivo (Ilustración 4.14) para que el lectorista pueda abrir el gabinete sin disparar una alarma de seguridad programada.

Ilustración 4.15 Interfaz para deshabilitar el interruptor de puerta.

En el evento de que el oficinista desee deshabilitar el switch de la puerta para que el personal de campo pueda tener acceso al interior del gabinete contenedor de medidores se ingresa el identificador del gabinete dentro del segmento de red específico.



Ilustración 4.16 Lista de tareas para la sección de Usuarios.

La siguiente sección dentro del menú del Head End es la que presenta las operaciones a realizar con los servicios pertenecientes a los usuarios dentro de la red eléctrica como se observa en la Ilustración 4.16.



Ilustración 4.17 Interfaz para ingreso de datos para agregar un nuevo usuario.

La operación de agregar un nuevo usuario al sistema desde la interfaz del Head End (ilustración 4.17), se comienza cuando el personal de oficina ingresa los identificadores de los dispositivos asociados al servicio dentro de cajas de texto para que este sea dado de alta dentro de la base de datos al presionar el botón “agregar usuario”.



Ilustración 4.18 Lista de tareas para servicios.

Los servicios también pueden ser conectados y desconectados desde la interfaz web del sistema ingresando al submenú de Servicios dentro del menú de usuarios como se muestra en la ilustración 4.18.

Las opciones para conectar y desconectar un servicio presentes en el submenú de servicio mostradas en las ilustraciones 4.19 y 4.20 llevan al usuario a navegar dentro del Head End hasta las interfaces donde se presenta una caja de texto para introducir el Registro permanente de usuario o RPU (Ilustraciones 4.21 y 4.22) que sirve como identificador único de cada servicio dado de alta dentro de los sistemas de la comisión federal de electricidad.



Ilustración 4.19 Conexión de un nuevo servicio.



Ilustración 4.20 Desconexión de servicio.



Ilustración 4.21 Interfaz para ingreso de datos para conectar a un usuario.



Ilustración 4.22 Interfaz para ingreso de datos para desconexión de usuarios.



Ilustración 4.23 Eliminación de usuarios.



Ilustración 4.24 Interfaz para eliminar usuarios.

En cuanto a la eliminación de usuarios del sistema, se presenta el submenú de eliminar dentro del cual el usuario debe de introducir el RPU del servicio que desea dar de baja, las interfaces mediante la cual el oficinista podrá realizar estas tareas se muestran en las ilustraciones 4.23 y 4.24.



Ilustración 4.25 Lista de tareas para el menú de usuarios de Hand Held (Terminales Portátiles).

El siguiente menú presente en la interfaz web del Head End es llamado usuario de Hand Held (ilustración 4.25), este tiene como finalidad ser un medio de comunicación con las terminales portátiles usadas en campo por los lectores para realizar el registro de las actividades desempeñadas en campo.



Ilustración 4.26 Interfaz para introducir los datos de alta de usuario.

El oficinista puede dar de alta a un usuario para las terminales portátiles al ingresar en la sección de Agregar usuario Hand Held e ingresando los datos del personal al que será asignado el dispositivo como se puede observar en la ilustración 4.26.



Ilustración 4.27 Selección de tipo de rol de usuario.

Al agregar un usuario para la Hand Held, el oficinista puede seleccionar de entre todos los tipos de usuario que fueron dados de alta para poder hacer uso de este tipo de dispositivos como se muestra en la ilustración 4.27, lo cual se verá reflejado en el tipo de acceso a la aplicación de la terminal portátil con la cual realiza sus registros.

La siguiente funcionalidad del sistema es la de cambiar, la cual permite al oficinista editar los datos relacionados con los usuarios de las terminales portátiles como se muestra en la lista desplegable de la ilustración 4.28.



Ilustración 4.28 Lista de tareas para edición de información de usuario.

En las ilustraciones 4.29, 4.30, 4.31, 4.32 se puede observar las diferentes interfaces brindadas por el sistema para poder ingresar los parámetros a editar correspondientes a los usuarios de las terminales portátiles dadas de alta dentro del Head End.

Ilustración 4.29 Interfaz para la edición de nombre de usuario.

Ilustración 4.30 Interfaz para la edición de contraseña de usuario.



Ilustración 4.31 Interfaz para cambio de nombre de usuario.

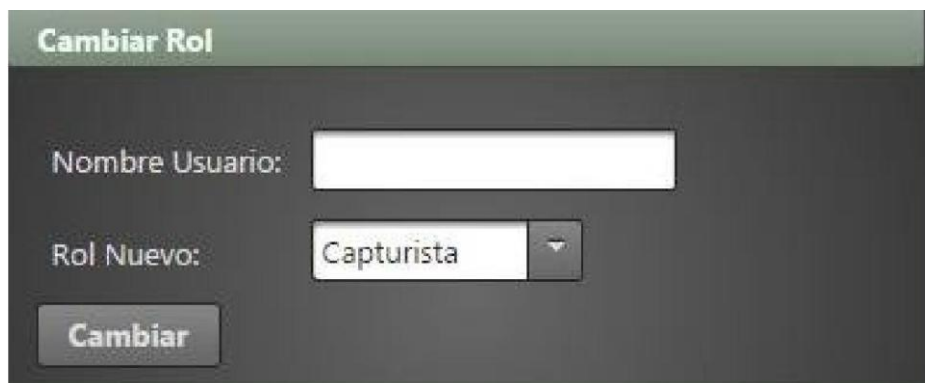


Ilustración 4.32 Interfaz para edición de rol de usuario.



Ilustración 4.33 Eliminación de un usuario identificado por nombre de usuario.

Los usuarios agregados en la base de datos para tener acceso a las terminales portátiles también pueden ser removidos de esta accedendo a la sección de “Eliminación de usuario” dentro del menú de usuarios de las Hand Held e ingresando el nombre único de usuario con el que este tiene acceso a los dispositivos; todo esto se realiza a través de la interfaz mostrada en la ilustración 4.33.

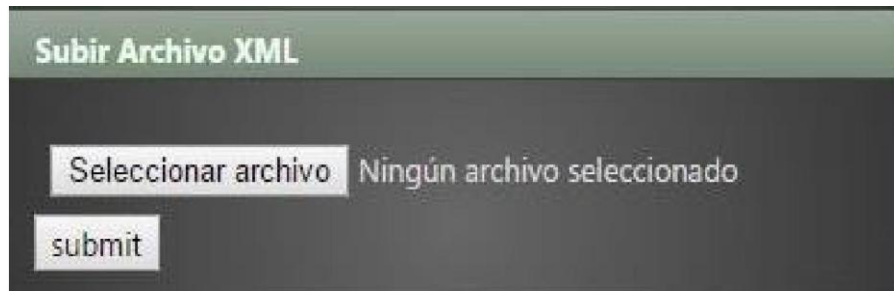


Ilustración 4.34 Selección de archivo para configuración.

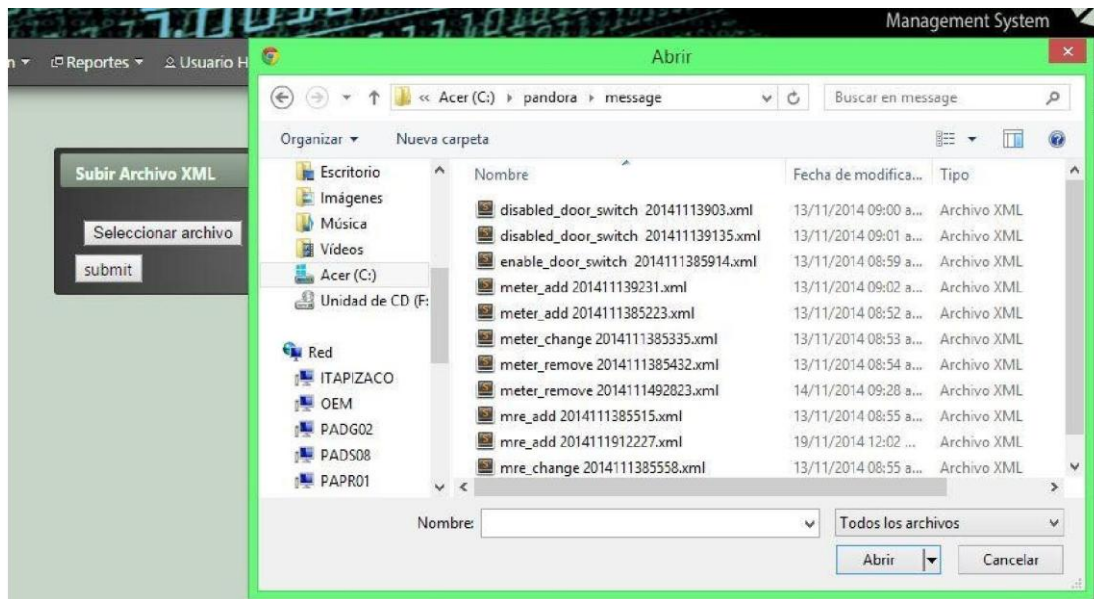


Ilustración 4.35 Cuadro de dialogo para seleccionar el origen del archivo.

Por último, se encuentra dentro de esta sección el submenú para enviar archivos XML a las terminales portátiles, estos archivos contienen las configuraciones para el correcto funcionamiento de la aplicación móvil instalada en las terminales mediante la cual el lectorista podrá realizar el registro de sus tareas desempeñadas.

El sistema mostrará un cuadro de dialogo de navegación de directorios para que el usuario del sistema pueda seleccionar el origen del archivo a ser enviado hacia el dispositivo seleccionado (ilustraciones 4.34 y 4.35).



Ilustración 4.36 Lista de tareas para variables de configuración.

Dentro del menú de variables de configuración se encuentran las diferentes tareas que pueden ser manipuladas desde la interfaz para modificar el funcionamiento de los dispositivos dentro de la red inteligente (ilustración 4.37).

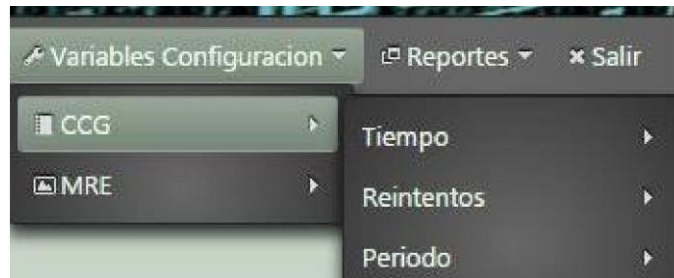


Ilustración 4.37 Lista de tareas para variables de configuración de CCG.



Ilustración 4.38 Lista de tareas para configurar tiempos de CCG.

La primer configuración a la cual puede acceder el oficinista es la de tiempo de expiración de PLC, IR y EIR, puertos de flujo de datos dentro del dispositivo encargado de guardar las configuraciones de los dispositivos dentro del gabinete de medidores, como puede observarse en la ilustración 4.38.



Ilustración 4.39 Interfaz para edición de tiempo de PLC.

Dentro de la sección de configuración de tiempo se mostrará primeramente el tiempo establecido actual y se brindara un campo para poder ingresar el nuevo tiempo a establecer, como se muestra en la ilustración 4.39.



Ilustración 4.40 Lista de tareas para reintentos de CCG.

Los reintentos mencionados dentro de la sección de CCG, se refieren al número de veces que el dispositivo tratará de solicitar información por los diferentes puertos de comunicación por lo cual como se puede observar en la ilustración 4.40 se despliega una lista de elementos referentes a los puertos de la caja de control.



Ilustración 4.41 Lista de tareas para configuración de periodos de CCG.

Los periodos desplegados a manera de lista, son tiempos configurables para la realización de tareas por parte de los dispositivos dentro del gabinete de medidores, como cajas de control, medidores y MREs enlazados. La lista de estos periodos a ser configurados es mostrada en la ilustración 4.41.



Ilustración 4.42 Lista de tareas para MRE.

Los módulos remotos de energía o dispositivos mediante los cuales el usuario es capaz de observar un estimado de consumo de energía también pueden ser configurados desde el Head End ingresando al menú de variables de configuración. Ilustración 4.42.



Ilustración 4.43 Lista de tareas para configuración de tiempos de MRE.

La primer configuración para el modulo remoto de energía, mostrada en la ilustración 4.43, se refiere a los tiempos en que este ejecutara sus tareas o bien escuchara por una instrucción. De este modo al cargar estas configuraciones al dispositivo este puede ejecutar sus tareas con una interacción mínima con los lecturistas; lo cual era uno de los objetivos de este desarrollo.



Ilustración 4.44 Lista de tareas para configuración de tiempos de pantalla de MRE.

La lista de las configuraciones de tiempo propias del MRE es mostrada en la ilustración 4.44 donde se observan las opciones de cambio para la pantalla del dispositivo a través de la navegación que llevara al usuario del sistema hasta la tarea que desea configurar para que esta sea enviada a través de la red de medidores hasta que llegue a uno o más dispositivos según sea el caso.



Ilustración 4.45 Lista de tareas para reportes de alarmas.

Finalmente dentro de las secciones del menú se encuentra una sección para la generación de reportes estadísticos de las funcionalidades del sistema en general, lo cual brinda a la comisión información acerca de eventos y fallas dentro de la red de medidores inteligente (ilustración 4.45).



Ilustración 4.46 Pantalla de Historial de variables de configuración.

Otra de las funcionalidades que brinda información que compete a la empresa desarrolladora del proyecto es la sección para generar un historial del cambio en las configuraciones de los dispositivos dentro de la red, cuya interfaz se muestra en la ilustración 4.46.

ID	Dispositivos	Anterior	Actual
1	CCG	60000	80000
2	CCG	80000	12000
3	CCG	6	7
4	CCG	80000	90000
5	CCG	2	3
6	CCG	15000	80000
7	CCG	4	5
8	CCG	90000	70000
9	CCG	70000	80000
10	CCG	80000	90000

Ilustración 4.47 Interfaz de consulta de historial de variables de configuración.

Los reportes generados para los dispositivos pueden ser exportados en archivos con formato PDF desde la interfaz mostrada en la ilustración 4.47, para poder generar un registro histórico del uso del sistema con respecto a estas variables así como la disminución de registros mostrados al usuario al momento de la consulta.

en las cuales se evalúan aspectos internos del sistema desarrollado. Para su realización, se seleccionaron una serie de tareas que tuvieron como finalidad analizar cada módulo que integra el sistema para comprobar que efectivamente los datos que fluyen a través de sus funciones y métodos se envían a través de estos de manera correcta y es procesada acorde a las necesidades planteadas por el usuario final.

Las pruebas realizadas para la caja negra incluyeron una serie de tareas ejecutadas por otro grupo de integrantes del equipo desarrollador donde se buscó principalmente evaluar entradas y salidas de datos al igual que la interacción como el sistema sin considerar el procesamiento de la información.

Debido a la naturaleza del sistema informático desarrollado, que cuenta con una arquitectura cliente – servidor se evaluaron aspectos críticos de una aplicación web como navegación, acceso, seguridad, etc. Así como validaciones de información ingresada y presentada en la interfaz del sistema.

El objetivo de la realización de todas estas pruebas es garantizar que el sistema realizara las tareas para las que fue diseñado de manera correcta, generando cálculos, enviando y recuperando registros, generando registros de eventos y creando archivos y documentos contenedores de datos correctos; lo cual garantiza un desarrollo de calidad que brinde satisfacción al cliente tanto funcional como visual, presentando una interfaz fácil de comprender y explorar para nuevos usuarios, siendo intuitiva y amigable.

La Tabla 4.1 muestra la lista de pruebas realizadas por el equipo programador donde se evalúan los aspectos principales de una aplicación web enfocadas a conceptos de cajas blanca y negra.

Tabla 4.1 Registro de pruebas de funcionalidad y desempeño

Nombre de la prueba	Resultado
Caja de vidrio	
Modularidad del sistema (secciones de funcionalidad).	√
Coherencia de variables y constantes.	√
Modelado de tablas.	√
Creación de entidades.	√
Conexión con la base de datos.	√
Inserción de registros.	√

Edición de registros.	√
Borrado de registros.	√
Registro de eventos.	√
Modelado de funciones y metodos.	√
Procesamiento de datos (operación de variables y constantes)	√
Envío de parametros entre funciones.	√
Creación de archivos XML.	√
Concatenación de datos para archivos XML.	√
Conexión con dispositivos HH.	√
Exportacion a documento PDF	√
Concatenación de datos para archivo PDF	√
Caja negra	
Congruencia de diseño de interfaz	√
Tamaño de texto	√
Validación de campos (tipos de datos a ingresar, tamaños, núlos)	√
Presentación de mensajes de ejecución de tareas.	√

La realización de las pruebas dentro de la sección de caja de vidrio de la Tabla 4.1 brindó una retroalimentación al equipo de programadores a lo largo de todo el proyecto ya que cada prueba se ejecutó tan pronto como se terminaba la fase de programación de cada módulo que conforma el sistema. De este modo se comprobó que los datos ingresados desde la interfaz fueran operados de manera correcta generando informacion clara al usuario final, garantizando que cada parte terminada de manera individual cumpliera con su cometido de manera satisfactoria.

Cada sprint programado durante el desarrollo se presentaba el adelanto logrado por el equipo donde se realizaban pruebas rapidas a la funcionalidad frente al usuario para demostrar los resultados obtenidos durante el tiempo entre sprints y la fiabilidad de estos, por lo cual cada entrega se registraba el avance alcanzado el documento de back log presentado en el Anexo B.

Una vez terminada la presentación el equipo desarrollador realizaba un registro de eventualidades que incluía fallas observadas por los integrantes que asistían al Sprint, así como faltas o inconsistencias mencionadas por el usuario final para poder ser atacadas de inmediato corrigiéndolas en momentos tempranos y oportunos, evitando correcciones mayores que implicarían uso de recursos como tiempo y esfuerzo en la entrega final del proyecto.

Tras esto se comenzó la fase de integración de funcionalidad donde se evaluaron los aspectos contenidos dentro de la sección de caja negra que prueban principalmente la fluidez del sistema en cuanto a uso e interactividad.

Como parte de el convenio realizado entre el Instituto Tecnológico de Apizaco y la empresa Softek Global eDesign se expidió una carta de satisfacción del cliente avalando el trabajo descrito por el autor del presente documento, la cual se encuentra integrada en el anexo .



"2015, Año del Generalísimo José María Morelos y Pavón"

TECNOLÓGICO NACIONAL DE MÉXICO
Instituto Tecnológico de Apizaco



CAPITULO V CONCLUSIONES Y TRABAJOS FUTUROS

CONCLUSIONES

El desarrollo del proyecto descrito en el presente documento cumplió con las expectativas de desempeño del producto final entregado a la empresa Softtek Global eDesign quedando como registro la carta de satisfacción del cliente incluida en la sección de Anexos de esta tesis.

El proyecto brinda un método de enlace con las terminales portátiles que mantienen el registro de operaciones realizadas en campo por el personal lectorista durante su jornada de trabajo, enviando un respaldo de esta información hacia los destinos correspondientes; es decir, bases de datos y sistemas informáticos que mantendrán un resguardo de la información recabada. El servicio de Windows descrito en la sección 3.11 detecta la conexión de este tipo de dispositivos para obtener el archivo que contiene la información de las tareas ejecutadas analiza el documento y se comunica con los Head Ends encargados de distribuir la información ya sea hacia los sistemas de CFE o bien a la base de datos de NUBO o PANDORA según corresponda. Durante el proceso de desarrollo se analizaron una serie de técnicas para análisis de documentos mencionadas en el capítulo 1.8 de este documento. Se optó por métodos rápidos y confiables basados en árboles DOM de entre los analizados en la sección 2.2 para la lectura de estos documentos para hacer más eficiente la aplicación, el hecho de que esta esté basada en un Windows Service implica que no habrá una interfaz gráfica con la cual interactuaría el usuario del sistema, pero brinda la ventaja de hacer el funcionamiento transparente para el oficinista. La problemática de generación de reportes de errores fue resuelta con la creación de un archivo de registro de acontecimientos que el Head End se encarga de analizar para realizar una operación que brinde una solución o envíe las alertas necesarias a los administradores del sistema.

La implementación de metodologías ágiles para el desarrollo de este módulo mencionadas en el capítulo 2.3 brindó grandes ventajas en cuanto a rapidez de programación y entendimiento de la funcionalidad esperada para el servicio. Las tareas fueron correctamente asignadas y distribuidas dentro del equipo de trabajo gracias a un correcto análisis y planeación de actividades.

El módulo Web (sección 3.11) que está encargado de brindar una interfaz a la cual puede acceder el oficinista de la comisión de manera remota, implicó una mayor

complejidad al igual que un mayor esfuerzo por parte de los integrantes del equipo desarrollador.

Para comenzar con el desarrollo de esta funcionalidad se comenzó con una fase prototipado de Servlets y Java Server Pages para el entendimiento más concreto del flujo de información y el comportamiento del sistema. Para poder comenzar la interacción de este con la base de datos se realizaron estudios para la selección de la mejora alternativa de entre los proveedores de bases de datos con mayor prestigio.

La conclusión obtenida de estos estudios fue basada en precio, seguridad, espacios de almacenamiento y facilidad de manejo, arrojando como mejor candidato a ORACLE, un gestor de base de datos poderoso, robusto y con grandes espacios de almacenamiento acordes a las necesidades del usuario así como un sistema de seguridad confiable (capítulo 3.5).

El modelado de la base de datos descrito en la sección 3.6 surgió a partir de las entidades pertenecientes a la nueva red de medidores inteligente y los identificadores que sirven para la interacción entre estos para generar finalmente información útil para los procesos de facturación de la comisión federal de electricidad. Se seleccionaron cuidadosamente los campos indispensables para representar cada entidad así como las relaciones entre estas para generar las dependencias de los dispositivos a un usuario. Con el paso del tiempo la base de datos fue creciendo al agregar nuevas tablas y campos según fueron surgiendo las necesidades y nuevos desarrollos en el proyecto general. El resultado, descrito a lo largo del capítulo 4 de este documento presenta una base de datos contenedora de información referente a los nuevos dispositivos y sus dependencias ya que la mayor parte de los datos generados ya son almacenados actualmente en las bases de datos de NUBO y CFE.

Debido a la duplicidad de información entre las bases de datos de los sistemas que coexisten en interacción con Pandora descrita en la sección 3.6, se escogieron solamente los datos referentes a los dispositivos que no son almacenados en las bases de datos de CFE, lo cual implica el establecimiento de un servidor con las capacidades suficientes para guardar la cantidad de registros planeada así como poder expandirse sin problemas en un futuro cercano conforme el sistema sea implementado a través del país.

Una vez establecida la base de datos del modo descrito en la sección 3.6 del presente documento, se comenzó con el modelado de la interfaz web presentando las funcionalidades básicas para la primera fase de desarrollo.

Durante cada sprint se analizó una lista de tareas que el sistema debería de realizar, se le asignaron prioridades a cada una de estas así como analizar cada una de ellas para establecer su viabilidad y utilidad. Cada avance entregado era registrado en los documentos de BackLog que forman parte del Anexo B de este documento, registrando el nivel de completitud a la fecha del Sprint para reflejar el alcance logrado.

La asignación de tareas por parte del jefe de proyecto fue basada en las habilidades de cada uno de los integrantes así que cada integrante tenía una carga de trabajo diferente, presentando una mayor inclinación hacia el jefe de proyecto y el autor de este documento los cuales se encargaron de el manejo del proyecto e incursionaron del mismo modo en la programación de las bases del sistema. Incluso se implementó programación extrema para la integración de funcionalidad desarrollada por integrantes del equipo ubicados en diferentes locaciones geográficas durante la fase del inicio del desarrollo del módulo web descrito en la sección 3.10.

Se estableció usar una nomenclatura para las variables que representara su funcionalidad del sistema con nombres en inglés, cada método y clase fue documentada en código también en inglés para fácil entendimiento y adaptación en caso de que nuevos integrantes fueran adheridos al proyecto.

Basados en técnicas de desarrollo de software se realizaron los diagramas que describen el flujo de información, módulos e interacción de funciones que conforman el Head End (secciones 3.7, 3.8, 3.9). Estos diagramas sirven para que cualquier persona pueda entender el funcionamiento del sistema en diferentes niveles de abstracción, de este modo se obtuvo un modelado de todas las partes que conformarían el sistema para asignar cargas de trabajo y prioridades a cada sección planeada.

Una vez establecidos los módulos que conforman el Head End se tuvo que considerar las interacciones que este tendría con otros sistemas informáticos descritas en la sección 1.2, para poder elegir la mejor manera en que esta comunicación se desempeñara de la manera más rápida y eficaz.

Debido a que el web Service que comunicará al Head End con NUBO descrito en la sección 5.1 de este documento, el lenguaje de programación seleccionado para la interfaz web fue Java, este lenguaje cuenta con plug ins que facilitan el trabajo con aplicaciones Web, este es el JavaEE (Java Enterprise Edition) el cual fue descargado sin costo a través del menú de plug ins del IDE Net Beans, aplicación seleccionada para el desarrollo del proyecto.

Para el modelado de las entidades de la base de datos dentro del sistema se utilizó el Framework Hibernate el cual como se mencionó previamente en la sección 1.8 de este documento se encarga de modelar los objetos de datos y java Beans de cada una de las tablas para facilitar el manejo de esta información para el desarrollador.

Prime faces, un Framework también gratuito fue utilizado para la creación de la interfaz de usuario ya que brinda controles vistosos fáciles de implementar en aplicaciones Web como se describe en la sección 2.4. Esto eliminó la implementación de tiempo de diseño y prototipado de estas interfaces por parte del equipo desarrollador ya que de este modo el mayor esfuerzo fue usado para adaptar solamente en su totalidad los menús y formularios del sistema como se muestra en la sección 4.10.1. El resultado de este desarrollo es mostrado y descrito a profundidad a lo largo del capítulo 4 de este documento.

Por ultimo podemos concluir se ha comprobado que es posible el desarrollo de un sistema informático de tipo Head End aplicando metodologías ágiles y técnicas de programación como Extreme Programming y Modular Programming gracias a los análisis y registros presentados a lo largo del documento, quedando como mayor prueba de esto la carta de satisfacción del cliente que se incluye en el Anexo Bde esta tesis. Este sistema será puesto a prueba en un ambiente real una vez que la red inteligente comience a ser instalada en las locaciones geográficas seleccionadas como pruebas piloto.

TRABAJOS FUTUROS

Como trabajos futuros se establecieron las pruebas de un socket TCP IP desarrollado en java el cual debe de formar los mensajes generados desde el modulo web para poder ser enviados a través de la red de medidores y ser interpretados por estos.

Este mensaje deberá de ser enviado al colector de cada red que corresponde a una central de CFE, este colector, basándose en sus tablas de rutas deberá de encaminar el mensaje para que este llegue a su destino.

Para la comunicación con el Head End NUBO hace uso de un Web Service programado en Java, NUBO es un sistema que brinda acceso a los sistemas de CFE por lo cual cualquier aplicación desarrollada para la comisión deberá primero comunicarse con NUBO y este se encargara de enviar la información hacia SICOSS y SICOM según corresponda.



TECNOLÓGICO NACIONAL DE MÉXICO
Instituto Tecnológico de Apizaco

"2015, Año del Generalísimo José María Morelos y Pavón"



Cabe resaltar que la fase de compatibilidad quedó en espera durante el proceso de redacción del presente documento, en esta fase se deberá analizar el funcionamiento del servicio con las diferentes marcas de dispositivos con las que trabaja la comisión federal de electricidad.

REFERENCIAS BIBLIOGRAFICAS

- Adam Dukovich, J. H. (2008). JOXM: Java Object —XML Mapping. ieeexplore.ieee.org, 4.
- Barton, B. (2009). All-Out Organizational Scrum as an Innovation Value Chain. <http://ieeexplore.ieee.org/>.
- Beck, K., & Fowler, M. (2000). Planning Extreme Programming. Addison-Wesley Professional.
- Brown, S., Dalto, S., & Jepp., D. (2011). Pro JSP 2. Springer.
- Carl Binding, D. B.-W. (2000). Generation of Java Beans to Access XML Data. ieeexplore.ieee.org, 7.
- Christian Kirkegaard, A. M. (2004). Static Analysis of XML Transformations in Java. ieeexplore.ieee.org, 16.
- Chromatic. (2013). Extreme Programming Pocket Guide. O'Reilly Media.
- Dong, X., Hongxing, W., Tianmiao, W., & Suibing, Z. (2012). A smart metering system for monitoring electricity of building based on wireless network. ieeexplore.ieee.org, 6.
- Elliott, J. (2004). Hibernate: A Developer's Notebook. O'Reilly Media, Inc.
- Englander, R. (1997). developing Java Beans. O'Reilly.
- EROL-KANTARCI, M., & MOUFTAH, H. T. (2012). SURESENSE: SUSTAINABLE WIRELESS RECHARGEABLE SENSOR NETWORKS FOR THE SMART GRID. ieeexplore.ieee.org, 7.
- Fan, R., Cheded, L., & Toker, O. (2005). Java plus XML: a powerful new combination for SCADA systems. <http://ieeexplore.ieee.org/>.
- Goel, A. (2010). Computer Fundamentals. Pearson Education India.
- Hlvats, I. (2013). Instant PrimeFaces Starter. Packt Publishing.



TECNOLÓGICO NACIONAL DE MÉXICO
Instituto Tecnológico de Apizaco

"2015. Año del Generalísimo José María Morelos y Pavón"



- Hunt, J. (2012). *Essntial Javabeans fast*. Springer Science & Business Media.
- Hunter Jason, C. W. (2001). *Java Servlet Programming*. O'Reilly.
- Inc., K. L. (2010). *C# Programming,Covers .NET3.5, Black Book, Begginers* Ed.DreamTEch Press.
- Institute, P. M. (2009). *Guia de los Fundamentos Para la Direccion de Proyectos (Guia del PMBOK)* .Project Management Institute.
- Jixuan Zheng, L. L. (2013). *Smart Meters in Smart Grid: An Overview*.
ieeexplore.ieee.org, 8.
- Kenneth C. Budka, J. G. (2014). *Communicarion networks for Smart Grids: Making SmartGrid Real*. Springer.
- Kent Beck, C. A. (2005). *Extreme Programming Explained embrace Change*. Pearson Education.
- Khalane, T., & Tanner, M. (2013). *Software quality assurance in Scrum: The need for concrete guidance on SQA strategies in meeting user expectations*.
<http://ieeexplore.ieee.org/>.
- Layka, V. (2014). *Learn Java for web development*.Apress.
- Lopez, J. E. (2015). *Diseño e implementación de un módulo de comunicación basado en radiofrecuencia y protocolo 6LowPan para el envío de datos* diseño e implementacion de un modulo de comunicacion basado en radiofrecuencia y protocolo 6lowpan para el envio de datos . Instituto Tecnológico de Apizaco.
- Momoh, J. (2012). *Smart Grid: Fundamentals of design and analysis*. Wiley, IEEE Press.
- Mundra, A., Barclays, S., & Misra, S. (2013). *Practical Scrum-Scrum Team: Way to Produce Successful and Quality Software*. <http://ieeexplore.ieee.org/>.
- Palafox, M. d. (2015). *"Pandora Management System" Informe de residencias no publicado*. Instituto Tecnológico de Apizaco.
- Patterson, B. (2002). *Visual Basic .NET Windows Services Handbook*. Wrox.
- Patzer, A. (2002). *JSP Examples and best practices*. Apress.
- Pries, K. H., & Quigley, J. M. (2010). *Scrum Project Management*. CRC Press.
- Priolo, S. (s.f.). *Métodologias Agiles*.USERSHOP.



"2015. Año del Generalísimo José María Morelos y Pavón"

TECNOLÓGICO NACIONAL DE MÉXICO
Instituto Tecnológico de Apizaco



- Reddy, K. S. (2013). Primefaces Beginner's Guide. Packt Publishing.
- Robinson, S. (2004). Professional C#. John Wiley & Sons.
- Rubin, K. S. (2013). Essential SCRUM a practical guide to the most popular agile process. Pearson Education.
- Simonov, M. (2014). Hybrid Scheme of Electricity Metering in Smart Grid. ieeexplore.ieee.org, 8.
- Sutherland, J., & Patientkeeper, I. B. (2005). Future of scrum: parallel pipelining of sprints in complex projects. <http://ieeexplore.ieee.org/>.
- Syngress. (2002). Developing Web Services with Java APIs for XML Using WSDP. Syngress.
- Téllez, A. G. (2009). Learning Content Production with XML and Java. ieeexplore.ieee.org, 5.
- Yong, H. G. (2011). Study and practice of import Scrum agile software development. <http://ieeexplore.ieee.org/>.
- Zheng, J., Gao, D., & Lin, L. (2013). Smart Meters in Smart Grid: An Overview. Denver, CO: IEEE.



"2015, Año del Generalísimo José María Morelos y Pavón"

TECNOLÓGICO NACIONAL DE MÉXICO
Instituto Tecnológico de Apizaco



ANEXOS

A. CONCEPTOS Y ACRÓNIMOS.

Smart Metering: Es una tecnología diseñada para realizar mediciones del consumo en servicios públicos; los dispositivos encargados de realizar estas tareas serán capaces de desempeñarse de manera automatizada y con una mínima interacción con el usuario.

Smart Grid: Integración de una red de comunicación basada en una topología llamada Mesh Grid; los dispositivos que la integran tienen como característica una inteligencia que les permite formar y reformar dicha red de manera automática, además de formar de manera independiente rutas por las cuales se transportan los paquetes de información generada.

Auto Healing: cada dispositivo que forma parte de una red inteligente deberá de ser capaz de funcionar bajo esta tecnología; la cual permite reformar la red en caso de que ocurra una falla en cualquiera de los nodos formando de manera automática nuevas rutas para el flujo de la información.

Utility: sistema informático propiedad de la comisión prestadora del servicio público desde donde se generan las operaciones dirigidas hacia los dispositivos en la red inteligente; del mismo modo genera facturación y reportes.

Head End: Sistema informático encargado de manejar el flujo de información que va de la red de medidores hacia la Utility o viceversa.

GMM: Gabinete Modular de Medidores. Gabinete contenedor de los dispositivos de metrología; a través de él serán alimentados los servicios con energía para poder prestar el servicio de manera controlada.

MM: Medidor Modular. Dispositivo encargado de la medición del consumo de energía eléctrica, puede funcionar de manera automatizada y contiene sensores para diagnosticar su propio funcionamiento.

MRE: Módulo Remoto de Energía. Display (dispositivo con una pantalla) donde es presentada la información acerca del consumo de energía al cliente directamente en su hogar, pudiendo mostrar un estimado en pesos en tiempo real.

CCG: Caja de Control de Gabinete. Dispositivo electrónico encargado de controlar el flujo de información en el GMM entre los MM (será reemplazada por una mejorada).

PLC: Power Line Carrier. Tecnología diseñada para el envío de información entre dispositivos electrónicos haciendo uso de la línea de corriente eléctrica.

RF: Radio Frecuencia. Tecnología implementada en radios que integran una red de comunicación por la cual los paquetes de información fluyen de nodo a nodo hasta alcanzar su destino.

Windows Service: Aplicación de Software propia del sistema operativo Windows, se ejecuta característicamente en segundo plano y carece de una interfaz de usuario, su ejecución puede ser iniciada automáticamente una vez que se el sistema operativo arranca.

XML: eXtensible Markup Language, formato para documentos de texto que cuenta con una estructura a manera de tabla dividida por nodos, atributos y valores para cada campo en el cual se pueden modelar objetos de cualquier tipo.

CSV: Comma Separated Values, formato para documentos de texto que emula la estructura de una tabla, separando los valores en columnas por comas y las filas por un salto de línea dentro de su estructura.

Servlet: Aplicación desarrollada en el lenguaje de programación JAVA que se encuentra ejecutando se del lado del servidor dentro de una arquitectura Cliente – Servidor.

JSP: Paginas de Servidor de Java (Java Server Pages), son una tecnología desarrollada para poder programar páginas web con código java embebido.

Encriptación: Tecnología desarrollada para la protección de información en ambientes de tecnologías de la información proporcionando métodos de cifrado.

Redes AMI: infraestructura de Medición Avanzada (Advanced Metering Infrastructure), son sistemas que se encargan de la metrología dentro de una red de medidores inteligentes.



"2015, Año del Generalísimo José María Morelos y Pavón"

TECNOLÓGICO NACIONAL DE MÉXICO
Instituto Tecnológico de Apizaco



B. DOCUMENTO DE BACKLOG DE SPRINTS

Actividades y objetivos	Dueño	Plan de trabajo																				
		Abril	May	Jun	Jul	Ago	Sept	Oct	Nov	Dic												
PMS	Identificación de requerimientos del sistema.	ASA,C DG	█																			
	Establecer flujo de información	ASA,C DG	█																			
	Establecer métodos de entrada y salida de datos	ASA,C DG	█	█																		
	Establecer lenguajes de Programación	ASA,C DG	█																			
	Análisis y diseño del sistema (Diagramas)	ASA,C DG	█	█																		
	Planeación de funcionalidad esperada	ASA,C DG	█	█																		
		ASA,C DG																				
	Identificación de requerimientos para la Base de Datos	ASA,C DG	█	█																		
	Establecer un servidor de Base de Datos	ASA,C DG			█																	
	Modelado del diagrama Entidad Relación	ASA,C DG			█																	
	Establecer las relaciones entre las entidades	ASA,C DG			█																	
	Definir campos para la base de datos	ASA,C DG			█	█	█															



TECNOLÓGICO NACIONAL DE MÉXICO
Instituto Tecnológico de Apizaco

"2015, Año del Generalísimo José María Morelos y Pavón"





"2015. Año del Generalísimo José María Morelos y Pavón"

TECNOLÓGICO NACIONAL DE MÉXICO
Instituto Tecnológico de Apizaco



C. CONSTANCIA DE SATISFACCIÓN DEL CLIENTE



18 DE DICIEMBRE DE 2014

MTRO FELIPE ROSARIO AGUIRRE.
DIRECTOR DEL INSTITUTO TECNOLÓGICO DE APIZACO
PRESENTE

AT'N: M.A.D. Ma. A. ACELA DÁVILA JIMÉNEZ
ENCARGADA DE LA DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN

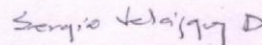
Por medio de la presente reciba un cordial saludo y al mismo tiempo hago constar que el M.C JOSÉ JUAN HERNÁNDEZ MORA, profesor investigador de la institución que usted dignamente dirige y el LIC. AGUSTÍN SÁNCHEZ ATONAL, alumno de la Maestría en Sistemas Computacionales con número de control M07370532, han asesorado y realizado servicio especializado en año 2014, en el desarrollo de su proyecto denominado:

"DESARROLLO DE UN HEAD END PARA LA GESTIÓN DE LA INFORMACIÓN DE UN SMART GRID", que consta en el diseño y desarrollo de un sistema integral de software de tipo Head End con aplicaciones desarrolladas en los lenguajes de programación de JAVA y C#.

La asesoría especializada y el servicio tecnológico realizado por el investigador y el alumno anteriormente mencionados para el desarrollo de este nuevo producto, cubren y han excedido las expectativas planteadas durante la fase de este proyecto. El acuerdo, desarrollo, seguimiento y resultados del mismo se encuentran a resguardo de nuestra empresa como secreto industrial.

Agradeciendo sus atenciones quedo de ustedes.

ATENTAMENTE



Sergio Velázquez Aguirre
Gerente de Operación





"2015. Año del Generalísimo José María Morelos y Pavón"

TECNOLÓGICO NACIONAL DE MÉXICO
Instituto Tecnológico de Apizaco



D. CONSTANCIA DE LIBERACIÓN DE ESTANCIAS.



CARTA DE LIBERACIÓN DE ESTANCIA PROFESIONAL

Querétaro, Querétaro a 18 de Julio de 2014.

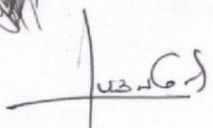
Asunto: Carta de Liberación de Estancia Profesional.

ING. FELIPE PASCUAL ROSARIO AGUIRRE
DIRECTOR DEL INSTITUTO TECNOLÓGICO DE APIZACO
PRESENTE:

Por medio de la presente me permito notificarle que el (la) alumno(a): **SÁNCHEZ ATONAL AGUSTÍN**, con número de matrícula: **M07370532** de la **MAESTRÍA EN SISTEMAS COMPUTACIONALES**, realizó de manera satisfactoria su **ESTANCIA PROFESIONAL** en esta Empresa con razón social: **SOFTEK GLOBAL EDESIGN SERVICES** ubicada en: **LUIS MALVAES No. 523, COL. REFORMA AGRARIA 2DA SECCIÓN QUERETARO, QRO.** Del día **20** del mes de **ENERO** al día **18** del mes de **JULIO** del año **2014** en el proyecto: **PROGRAMACIÓN Y CONFIGURACIÓN DE TRAJETA DE COMUNICACIÓN REMOTA PARA TRANSMISIÓN DE DATOS**, del área de **DESARROLLO DE SOFTWARE**, por lo cual se extiende su liberación.

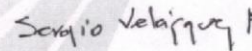
Recibi
original
6/Ago/14




ING. JUAN GONZÁLEZ SALOMÓN
GERENTE DE LA EMPRESA

ATENTAMENTE





ING. SERGIO VELÁZQUEZ AGUIRRE
ASESOR DE PROYECTO



"2015. Año del Generalísimo José María Morelos y Pavón"

TECNOLÓGICO NACIONAL DE MÉXICO
Instituto Tecnológico de Apizaco



E. PRODUCCIÓN ACADÉMICA

DESARROLLO DE UN HEAD END PARA GESTIONAR INFORMACIÓN EN UNA RED INTELIGENTE

Lic. Agustín Sánchez Atonal¹, M.C José Juan Hernández Mora², M.C. Ma. Guadalupe Medina Barrera³ M.C.
Yesenia Noemí González Meneses⁴

Resumen—En el presente artículo se describe el proceso de desarrollo para un sistema integral de tipo *Head End* y la integración de las diversas tecnologías implementadas para su elaboración, dicho sistema estará encargado de gestionar la información que fluye a través de una red inteligente integrando de este modo la red de dispositivos inteligentes con el sistema informático propio de CFE. El propósito del sistema es automatizar el proceso de recolección de información pertinente para la facturación del servicio eléctrico proporcionado, así como el envío de mensajes hacia la red inteligente para la asignación de tareas, ya sea programadas o de ejecución inmediata. El sistema integrará diversos lenguajes de programación, así como herramientas para establecer comunicación con la red de medidores inteligentes, dispositivos de tipo *Hand Held*, unificados para comunicarse con una interfaz web y una base de datos de Oracle.

Palabras clave—HEAD END, RED INTELIGENTE, MEDIDOR INTELIGENTE, SISTEMA INTEGRAL.

Introducción

Una red inteligente o *Smart Grid* es una red integrada por dispositivos “inteligentes” capaces de realizar sus tareas de manera automatizada con una interacción mínima con el usuario, siendo generalmente dispositivos para medición en el consumo de servicios (medidores inteligentes o *Smart Meters*, medidores remotos de energía, etc.)[1].

Este tipo de redes por característica deberá de ser capaz de formarse de manera automática integrando tantos dispositivos como su capacidad le permita o como sea configurada para aceptarlos.

Se hace referencia a una red de dispositivos con antenas de radio frecuencia que comenzarán con la formación de la red inteligente transmitiendo *beacon requests* o informes de su posición e identificación de dispositivo en una determinada frecuencia, para que dispositivos que se comuniquen a esta misma frecuencia, usando el mismo método de comunicación, sean capaces de entablar comunicación con el edge router o dispositivo encargado de la formación de esta.

En cuanto al flujo de la información dentro de la red inteligente, podemos encontrar un protocolo basado en la dirección MAC del dispositivo para poder hacer el enrutamiento de los paquetes y dirigirlos de esta manera al dispositivo correcto.

Un conjunto de archivos XML son creados haciendo uso de técnicas de acceso secuencial a archivos de texto desde el *Head End*, dentro del contenido del mensaje se encuentran la operación solicitada y los identificadores del dispositivo al que el mensaje va dirigido [2]. Los mensajes XML fluirán a través de la red ingresando por el coordinador siguiendo una ruta que este mismo genera en el momento de la generación de la red basándose en la calidad del enlace entre los dispositivos que la integran guardándola en su memoria y actualizándola cada vez que la red sea reformada.

A su vez los mensajes con la petición para la ejecución de la tarea son generados desde el sistema comercial de la comisión de electricidad empaquetados para un grupo de medidores. El *Head End* deberá de encargarse de la gestión del flujo de información para conectar la red inteligente con el sistema comercial

¹ Lic. Agustín Sánchez Atonal. Egresado del instituto Tecnológico de Apizaco, estudiante de la Maestría en Sistemas Computacionales del mismo. Agustin.sanchez.atonal@gmail.com (autor corresponsal).

² M.C. José Juan Hernández Mora Catedrático de Maestría en Sistemas Computacionales del instituto Tecnológico de Apizaco

³ M.C. Ma. Guadalupe Medina Barrea catedrática de Maestría en Sistemas Computacionales del Instituto Tecnológico de Apizaco.

⁴ M.C. Yesenia Noemí González Meneses catedrática de Maestría en Sistemas Computacionales del Instituto Tecnológico de Apizaco.

haciendo uso de los archivos XML alimentando una base de datos contenedora de las asociaciones de los nuevos dispositivos

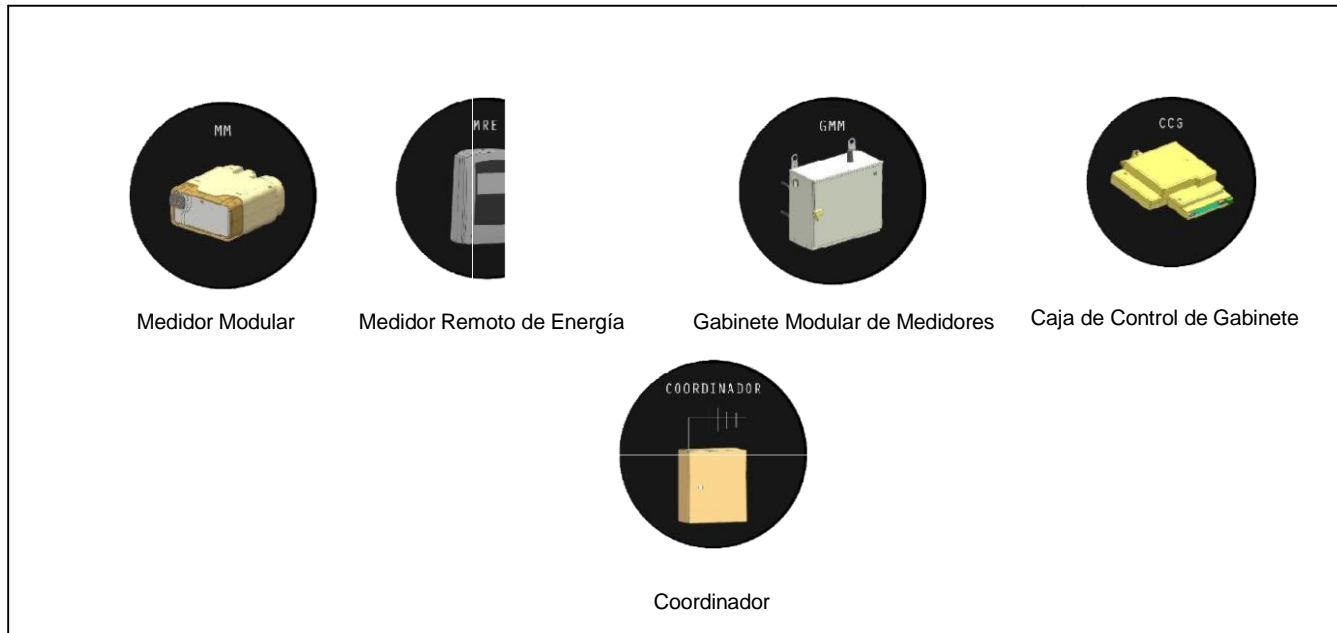


Fig.1. Dispositivos Que conformaran la red Inteligente.

En la figura 1 podemos observar los distintos dispositivos que conforman la red inteligente, el Medidor Modular (MM) encargado de los aspectos de metrología para el servicio, el Medidor Remoto de Energía o MRE encargado de mostrar en pantalla el consumo en Kw/h al igual que un estimado del consumo en pesos, el Gabinete Modular de Medidores (MRE) que concentra a los medidores, será ubicado en un poste con transformador en las calles, la Caja de Control de Gabinete (CCG) encargada de coleccionar la información de los medidores dentro de un gabinete, y finalmente el coordinador una tarjeta electrónica con funciones similares a las de una CCG pero con mayores capacidades de procesamiento y almacenamiento de datos; todos estos son los dispositivos a los que la aplicación deberá de enviar los mensajes contenedores de tareas en formato XML para que estos puedan ejecutarlas en el momento adecuado, además de recibir la información. La vía de comunicación se establecerá directamente con el coordinador y no directamente a los dispositivos; el coordinador cuenta con las capacidades necesarias para poder enrutar los paquetes al dispositivo adecuado.

Descripción del Método

Programación de un Windows Service

El sistema integral de tipo *Head End* recibe información de dispositivos denominados *Hand Held*, en estos dispositivos son asignadas las tareas al lectorista⁵ por parte de la comisión federal de electricidad, como por ejemplo dar de alta un servicio o bien, realizar la lectura de los medidores siguiendo una ruta específica. Al terminar con sus tareas, el lectorista habrá de guardar un registro de los resultados obtenidos por cada tarea guardando datos con los identificadores de los dispositivos, su estatus, sus asociaciones, etc. En la central de la comisión federal de electricidad el lectorista conecta vía USB el dispositivo para sincronizarlo con una computadora que tiene instalada la aplicación que recolecte la información del registro generado. Esta aplicación se desarrolló como un *Windows Service* programado en el lenguaje C#, que se ejecuta de manera transparente para el usuario, brindando facilidades como una mínima interacción con él y generando reportes de resultados de cada operación generada.

⁵ Término designado para los trabajadores de campo de la comisión federal de electricidad.

Un *Windows Service* es una aplicación propia del sistema operativo Windows que se ejecuta como un proceso en segundo plano para el usuario haciendo uso de la clase *server base*; es decir, la aplicación corre sin una interfaz gráfica minimizando la interacción con el usuario pero al mismo tiempo realizando todas las operaciones programadas de manera automática y precisa [3].

Esta aplicación está programada para ejecutarse al recibir informe de conexión de un dispositivo *Hand Held*. Actualmente en comisión federal de electricidad se utilizan varios modelos y marcas de *Hand Held* por lo cual la aplicación está desarrollada para poder funcionar con cada uno de ellos sin problemas.

Al ser conectado un dispositivo por el puerto USB del ordenador se desencadena un evento que envía mensajes a los servicios de ciertas aplicaciones, mensaje que será recibido por el *Windows Service* para poder acceder a los directorios propios de la *Hand Held* y realizando una búsqueda siguiendo una ruta de acceso en los documentos del dispositivo, se procederá a buscar por un archivo con extensión CSV en el que se encuentran los registros generados por las operaciones realizadas en campo. Dependiendo de la información obtenida por el proceso anterior en el archivo con extensión CSV la aplicación puede agregar en caso de un nuevo servicio o la sustitución de alguno de los dispositivos en la red inteligente registros nuevos dentro de tablas correspondientes hacia una base de datos de Oracle o bien actualizarlos usando una función programada de *UPSERT* en la cual se comprueba la existencia de los registros, comprobando un identificador único [4]. Como por ejemplo los números seriales de los dispositivos electrónicos y el RPU⁶ que la comisión federal de electricidad asigna para cada servicio; dentro de la funcionalidad también se desarrolló un módulo del sistema encargado de generar un archivo XML bajo un formato establecido comprensible para los dispositivos dentro de la red inteligente con un registro de tareas pendientes que no pudieron ser ejecutadas dentro de la operación en campo para ser enviadas al dispositivo al cual estas van dirigidas.

Desarrollo de una GUI usando Servlets y JSF

Para el desarrollo de la interfaz de usuario se implementó un *Servlet* de Java. Un *Servlet* es una aplicación que se ejecuta en el servidor aumentando su funcionalidad, facilitando al programador el desarrollo de aplicaciones web[5]. Dicho *Servlet* está enfocado a minimizar el consumo de recursos del lado del cliente facilitando el acceso al sistema, dejando de lado del servidor el peso de la ejecución de la aplicación. La implementación de la tecnología de JSF es descrita con simplicidad en [6].

El usuario accesa desde un URL al dominio contenedor del *Servlet* al que ingresa con un nombre de usuario y contraseña asignados. Dentro de este módulo del sistema integral puede realizar tareas de altas de servicios, sustitución de dispositivos, toma de lecturas instantáneas, programación de tareas como toma de lecturas e historiales de consumo, o hasta cortes de servicios. El menú principal se desarrolló usando *Java Server Faces* para presentar de manera amigable las funcionalidades programadas. En concreto se usará el *framework* llamado *Prime Faces* que brinda controles como bocetos sobre los cuales se puede trabajar adaptándolos a las necesidades.

Este módulo también crea archivos XML para ser enviados a través de la red inteligente hacia un dispositivo en específico y generará registros en la base de datos de Oracle según el resultado de la funcionalidad ejecuta

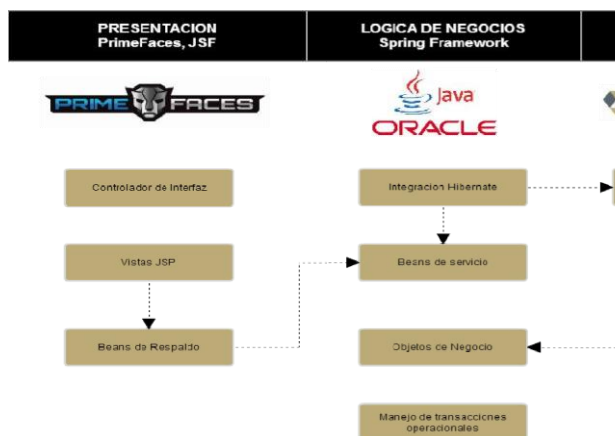


Fig.2. Estructura General de la Interfaz Gráfica de Usuario.

⁶ RPU Identificador asignado para cada servicio proporcionado por la comisión Federal de Electricidad (Registro Permanente de Usuario).

En la figura 2 se puede observar la integración de las distintas capas que integrarán la interfaz gráfica de usuario en sus distintos niveles de Front End, lógica de negocios y el Back End.

Aplicaciones para generar una cadena de conexión a la base de datos de Oracle

Por motivos de seguridad dentro del proyecto se optó por la utilización de entradas de registro de Windows donde se guardará de manera encriptada la cadena de conexión a la base de datos.

Debido a que los métodos de encriptación varían debido a los lenguajes de programación (Java para el *Servlet*, C# para el *Windows Service*) implementados en el desarrollo de la aplicación, se desarrolló una aplicación que generara la cadena de conexión propia de cada aplicación.

Desarrollo de un Socket programado en Java para la comunicación con la red inteligente

Este *socket* será el encargado de la comunicación de la interfaz web con el coordinador, para comenzar el proceso de enrutamiento de mensajes a través de la *Smart Grid*. El *socket* en cuestión será desarrollado en JAVA para generar homogeneidad con la interfaz web y facilitar de este modo la interacción entre los módulos.

Web Services Para La Comunicación Con El Head End NUBO

Estos serán programados en java, generados para ser consumidos enviando o recibiendo archivos XML dirigidos hacia los *SmartMeters*, en un directorio definido dentro del servidor remoto.

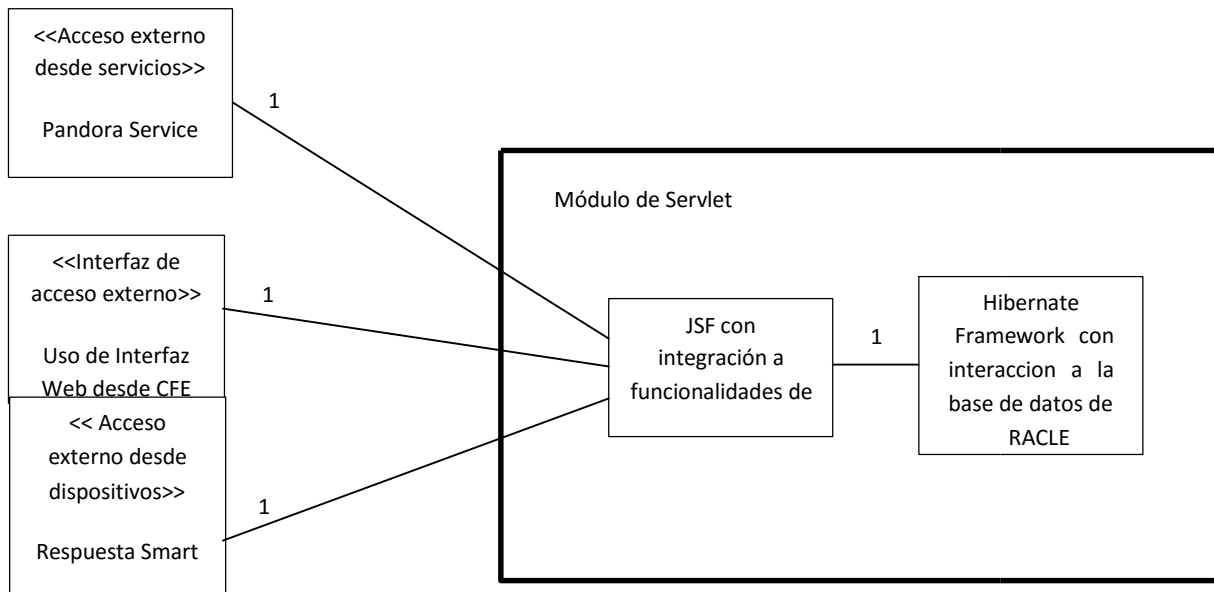


Fig.3. Diseño de arquitectura múltiples clientes – servidor con descripción de elementos.

Podemos observar además los diferentes medios de acceso al sistema como lo son, el *pandora Service* encargado de la conexión con los *Hand Held Devices* que interactuara con el *Servlet* en el núcleo de la aplicación enviando mensajes XML contenedores de la tarea a ejecutar por el sistema, el acceso haciendo uso de la interfaz web desde las terminales en la comisión que habrán de generar tareas y enviaran también archivos XML según la operación realizada, y finalmente una respuesta generada desde el *Smart Grid*.

El subsistema identificado en la ilustración como el *Servlet* habrá de interactuar con el *framework* de Hibernate, una descripción de cómo se comunica el *framework* con Oracle se describe en recibiendo los mensajes generados con los archivos XML indicándole la tarea a realizar con los registros de la Base de datos [7].

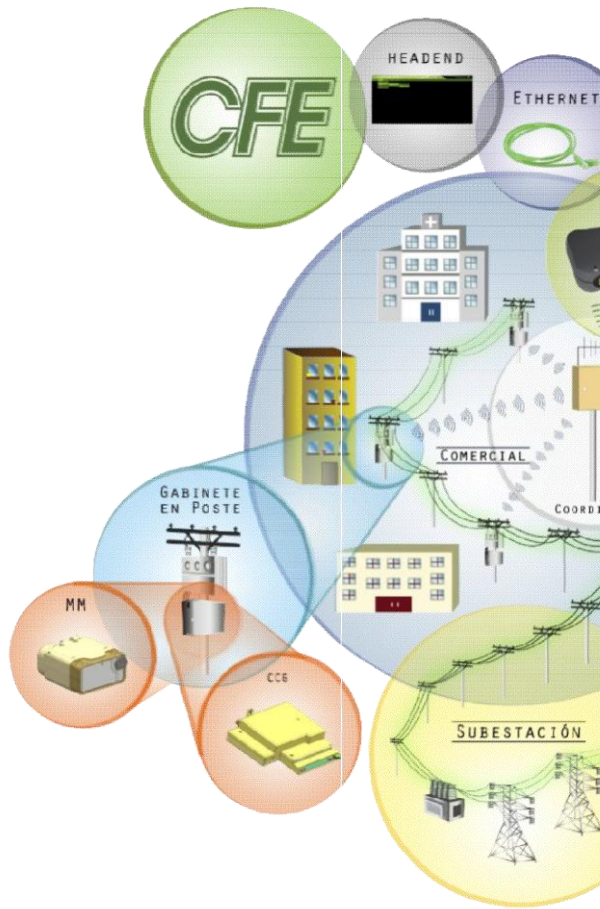


Fig.4. Esquema de la integración del Head End en la red inteligente.

En la figura número 4 podemos observar la ubicación del *Head End* dentro de la organización de la *Smart Grid* o red Inteligente. Ingresando desde la Comisión Federal de Electricidad el *Head End* proporcionará al usuario una interfaz amigable para que este pueda realizar las tareas deseadas tales como toma de lecturas, mediante el socket mencionado con anterioridad, este tendrá comunicación con el coordinador que tendrá a su cargo ya sea una red de servicios industriales o residenciales enviando paquetes a lo largo de la red hasta que estos alcancen su destino en el dispositivo correspondiente.

Las respuestas generadas por los dispositivos serán del mismo modo, siempre enviadas a través del coordinador de la red inteligente que, manteniendo un canal de comunicación bidireccional enviará al *HeadEnd* la respuesta para que este pueda informar al sistema integral homólogo NUBO, que es el encargado de la comunicación directa con el sistema informático de la comisión Federal de Electricidad para que el resultado de la operación quede asentado y registrado en los lugares correspondientes, como la base de datos propia del *Head End* en cuestión, en los registros de NUBO o bien en las bases de datos para el sistema de la comisión.

Resultados

Comunicación con dispositivos Hand Held	PENDIENTE
Generación de archivos XML	REALIZADO
Lectura de archivos XML	REALIZADO
Conexión del Sistema con la base de datos de Oracle	REALIZADO
Interacción con la base de datos implementando Hibernate	REALIZADO
Acceso desde la interfaz Web	REALIZADO
Funcionalidad de la interfaz web	REALIZADO
Pruebas de comunicación socket – Red de medidores	PENDIENTE

Tabla 1. Pruebas Realizadas para comprobar la funcionalidad del Head End.

En la tabla número 1 podemos observar un registro de las pruebas de funcionalidad que se realizaron para comprobar la correcta interacción de los módulos que forman parte del sistema así como la correcta ejecución de sus procesos y los resultados que estos generan, los módulos marcados con el estatus de pendiente serán descritos en trabajos posteriores debido a que la funcionalidad está siendo desarrollada actualmente.

Comentarios Finales

Finalmente cabe resaltar que la funcionalidad desarrollada para la comunicación con los dispositivos de tipo *Hand Held*, también interactuará con el *Head End* guardando registros a su base de datos o generando los archivos XML necesarios para que el sistema integral pueda recolectarlos de un directorio predefinido usando web services y ejecutar las tareas o guardar los registros debidos.

Conclusiones y trabajos futuros

El desarrollo del Sistema integral de tipo *Head End* brinda cierto grado de innovación para los procesos realizados cotidianamente dentro la Comisión Federal de Electricidad, automatizando la ejecución de estos con una mínima interacción del usuario, agilizando de este modo los procesos de facturación brindando información clara y precisa acerca de las mediciones de los servicios ya sea residenciales o industriales.

La integración de las tecnologías mencionadas garantiza una interacción confiable entre los módulos del sistema, así como la interacción con los dispositivos de la red inteligente o con los sistemas informáticos y *Head End* con los que este debe de comunicarse.

Las pruebas de funcionalidad para el modulo que realizará la comunicación con los dispositivos de tipo *Hand Held* y del módulo del socket para la comunicación con la red de medidores serán descritos en trabajos posteriores.

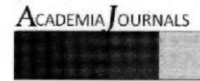
Referencias

- [1] Jixuan Zheng, Li Lin, David Wenzhog, "Smart Meters in Smart Grid: An Overview", *ieeexplore.ieee.org*, 2013.
- [2] Deitel Paul, Deitel Harvey "Java for Programmers", Prentice Hall, 2011.
- [3] Kogent Learning Solutios Inc., "C# Programming, Covers .NET3.5, Black Book, Begginers Ed.", DreamTEch Press, 2010.
- [4] Powell Gavin, "Oracle Data Warehouse Tuning for 10g", Elsevier Digital Press, 2005.
- [5] Hunter Jason, Crawford William, "Java Servlet Programming", O'Reilly, 2001.
- [6] B.M. Marwani, "Practical JSF Project Using Netbeans", PHI Learning, 2009.
- [7] H. Liu Henry, "Oracle Database Performance and Scalability: A Quantitative Approach", John Wiley & Sons, 2011.

Congreso Internacional de Investigación de **AcademiaJournals.com**
Celaya 2014



Certificado



Otorgado a

Lic. Agustín Sánchez Atonal

M.C José Juan Hernández Mora

M.C. Yesenia Nohemí González Meneses

M. en C. María Guadalupe Medina Barrera

por su ponencia intitulada

Desarrollo de un Head End para la gestión del flujo de información dentro de un Smart Grid

*la cual fue presentada en el Congreso Internacional de Investigación de Academia Journals
que se desarrolló los días 5 al 7 de noviembre de 2014
en las instalaciones del Instituto Tecnológico de Celaya,
Celaya, Guanajuato, México
y publicada en el portal de Internet*

Celaya.AcademiaJournals.com con ISSN 1946-5351 Online y 1948-2353 CD ROM



M.C. Miguel Ángel Melchor Navarro
Presidente de la Comisión Organizadora
Profesor de Ingeniería Industrial
Instituto Tecnológico de Celaya



Dr. Rafael Moras
Editor, AcademiaJournals.com
Profesor de Ing. Industrial y Administrativa
St. Mary's University, San Antonio, TX, EEUU

Y798 | Primer autor: Sánchez Atonal

Desarrollo de un sistema informático de tipo Head End para Redes Inteligentes

Lic. Agustín Sánchez Atonal⁷, M.C. José Juan Hernández Mora⁸,
M.C. Ma. Guadalupe Medina Barrera⁹ y Lic. Carlos Domínguez Galván¹⁰

Resumen— El presente artículo describe el desarrollo de un sistema integral de tipo Head End para el manejo de información producida dentro de una red inteligente de medidores para servicio eléctrico, el cual fue desarrollado dentro de la empresa Softek Global eDesign con la finalidad de automatizar el desempeño de tareas de facturación del servicio eléctrico integrando dispositivos de metrología y terminales portátiles para el registro de tareas de campo con un sistema informático mediante una interfaz web que interactúa a su vez directamente con una base de datos de ORACLE en la cual son registrados los resultados de las operaciones realizadas referentes al consumo del servicio y asociaciones de dispositivos con un servicio de hogar o industrial.

Palabras clave—Medidores Inteligentes, Red Inteligente, Head End, Sistema Informático, Servlet Java.

Introducción

Con el objetivo de brindar automatización a tareas desempeñadas dentro de comisión federal de electricidad se comienza con el desarrollo de un proyecto que logra impactar cada nivel la estructura de la red eléctrica actual.

La integración de nuevos medidores inteligentes capaces de realizar sus tareas de manera automatizada da inicio a la reformación de la red; estos nuevos medidores hacen uso de una red de radio frecuencia creando canales de comunicación de dos vías por los cuales podrán enviar los datos que estos mismos han recabado acerca de las mediciones obtenidas del servicio prestado; la comunicación entre los dispositivos físicamente se realiza bajo una topología de tipo Mesh Griddistribuida a lo largo de un área geográfica extensa limitada solamente por la intensidad de la señal para la comunicación y el tamaño predefinido para la tabla de direcciones dentro del colector.

Este último es un dispositivo que concentra la información recabada por todos los medidores inteligentes y los envía hacia los sistemas de información de CFE. Dentro de este canal de comunicación establecido entre la red inteligente de medidores y los sistemas informáticos de CFE debe de existir un sistema integral de tipo Head End encargado de manejar el flujo de información entre estas dos entidades. Dicho sistema deberá de ser capaz de comunicarse además con las terminales portátiles donde los trabajadores de campo registran los resultados de las tareas desempeñadas y brindar una interfaz de usuario para que los trabajadores de oficina programen tareas o bien soliciten información de la red de medidores en tiempo real.

⁷Lic. Agustín Sánchez Atonal estudiante de la Maestría en Sistemas Computacionales del Instituto Tecnológico de Apizaco. agustin.sanchez.atonal@gmail.com

⁸M.C José Juan Hernández Mora Catedrático de la Maestría en Sistemas Computacionales del Instituto tecnológico de Apizaco.

⁹M.C. Guadalupe Medina Barrera Catedrático de la Maestría en Sistemas Computacionales del Instituto tecnológico de Apizaco.

¹⁰Lic. Carlos Domínguez Área de Ingeniería de Software dentro de la empresa Softek global eDesign, jefe del presente proyecto.

Descripción del Método

Smart Grids (redes inteligentes) de electricidad.

Una Smart Grid es una red formada por el sistema entero de electricidad, el cual comprende la generación, transmisión y consumo de energía. Con respecto a la medición del consumo energético y envío de información para realizar tareas de facturación en la central de electricidad, se integran nuevos dispositivos denominados Smart Meters a la red, los cuales monitorean en tiempo real entregando información acerca de las estadísticas de consumo. (Zheng, Gao, & Lin, 2013)

La integración de estos nuevos dispositivos a la red eléctrica brinda un canal de comunicación de dos vías por el cual se pueden transmitir y recibir mensajes al mismo tiempo. De este modo los medidores pueden ser programados para realizar sus tareas de manera independiente de manera remota, desde las oficinas de CFE, o bien pueden enviar información para facturación en cualquier momento hacia la comisión.

Con el objetivo de manejar la información que se genera dentro de este canal se agrega a la estructura de la red inteligente un sistema integral de tipo informático denominado Head End; el cual será capaz de enlazar las ordenes y peticiones creadas desde la Comisión Federal de Electricidad con los medidores que forman parte de esta controlando de este modo el flujo de los datos producidos en tiempo real facilitando de este modo las tareas de facturación, corte y reconexión de un servicio industrial o de hogar.

Acerca de las metodologías empleadas para el desarrollo del proyecto

En cuanto a la metodología aplicada para el desarrollo del sistema integral informático se decidió implementar aquella con la cual se desempeñan los proyectos de manera cotidiana dentro de la empresa Softek Global eDesign; SCRUM es una metodología de desarrollo de software ágil. Debido a la naturaleza innovadora del desarrollo, los requerimientos de este fueron siempre evolucionando y mejorando, la adaptabilidad de esta metodología nos brinda rapidez y confiabilidad además de fácil adaptación a cambios estructurales permitiendo dividir el desarrollo del proyecto en fases establecidas de acuerdo a la priorización de los requerimientos encontrados haciendo entregas periódicas de avances funcionales. (Pries & Quigley, 2010)

Apegándonos a la técnica de programación del Modular Programming se realizó una descomposición del problema a atacar en módulos funcionales que además de ser independientes se relacionan una vez que el programa principal los llama para ejecución. Estos módulos están orientados en resolver tareas específicas para cada funcionalidad encontrada del sistema

completo enviando información pertinente de un módulo a otro; brindando facilidad y adaptabilidad en la fase de desarrollo debido al constante mejoramiento de los requerimientos establecidos. (Goel, 2010)

Se implementó un framework de desarrollo llamado Hibernatees un servicio de mapeo objeto relacional de Java el cual facilita el trabajar con bases de datos relacionales al crear objetos de Java (Elliott, 2004)

Para el caso exclusivo del presente desarrollo de trabajo particularmente con una base de datos de Oracle tras un estudio y análisis en relación a la cantidad de registros que se manejarán a futuro. De este modo los datos solicitados desde la base de datos son modelados en Java Beans a manera de estructuras contenedoras de datos para cada objeto solicitado creando métodos y atributos pertenecientes a la tabla modelada y sus campos.

Modularidad del sistema

La estructura general del sistema fue desfragmentada en módulos funcionales a un nivel en el cual cada módulo resolviera una problemática a la vez, lo cual facilitó la integración de nuevas funcionalidades o bien la adaptación de las funcionalidades ya existentes dentro del sistema cada vez que alguno de los requerimientos era cambiado.

Los dos principales módulos que comprenden la estructura del Head End son un Daemon desarrollado en C# que se encarga de la comunicación con las terminales portátiles; el otro es la interfaz web que se comunica a su vez con la red de medidores desarrollada usando tecnologías de Servlet de Java.

Integración con terminales portátiles

Para la fase de codificación del módulo de comunicación con las terminales portátiles comenzaremos describiendo la participación de estos dispositivos en la estructura de la red inteligente.

Una terminal portátil o Hand Held Device es un dispositivo inalámbrico con el cual son registrados los resultados de las operaciones realizadas por los agentes de campo; cuentan con una conexión de Infra rojo la cual proporciona comunicación con los medidores dentro del gabinete sin necesidad de abrirlo físicamente.

Una vez que el agente de campo vuelve de realizar sus tareas, este conecta la terminal portátil vía USB a un ordenador donde la información es descargada y valorada para poder ser registrada dentro de la base de datos. Si los registros fueron validados de manera exitosa, los cambios son guardados; pero en caso contrario, se registra en un LOG File las fallas generadas para que el sistema tome acciones pertinentes sobre estas.

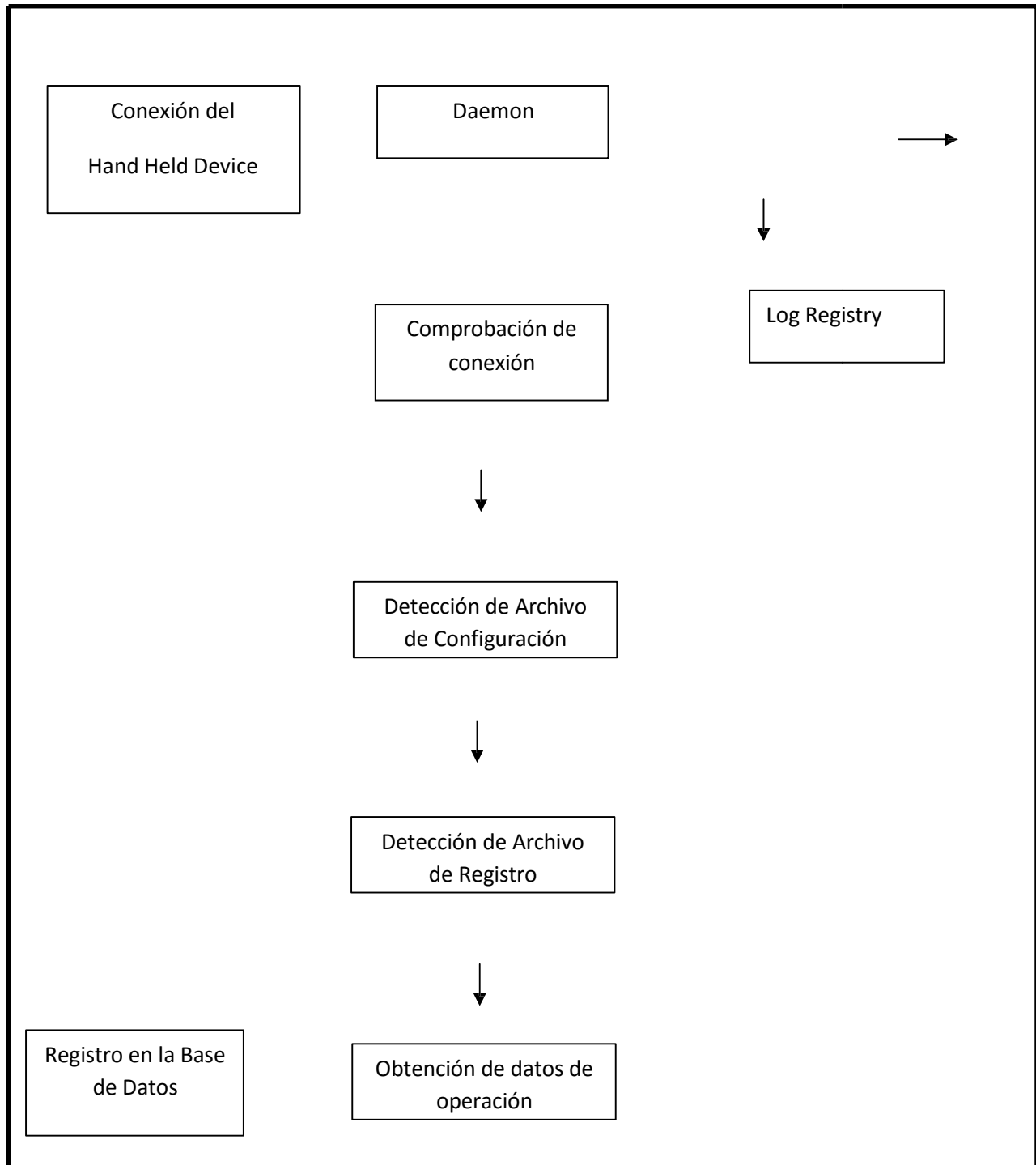


Fig.1. Diagrama a bloques de la estructura funcional del Módulo de Conexión con terminales portátiles.

Como se puede observar en la figura 1, el servicio se ejecuta en segundo plano sin una interfaz de usuario, lo cual permite que este funcione de manera automatizada dentro de un periodo de tiempo establecido. (Inc., 2010)

En busca de conexiones en los puertos USB del ordenador; una vez que la conexión es detectada, se procede a buscar dentro de los directorios de la terminal portátil por el archivo de registro de actividad, el cual contiene toda la información que el agente de campo guardo durante su jornada de trabajo. Este documento es interpretado por la misma funcionalidad del Daemon para obtener cada dato de manera individual, al mismo tiempo que registra los cambios si la operación fue realizada de manera exitosa. En el caso contrario crea una entrada en el archivo de registro con un código de error.

En caso de que hubiera ocurrido un error en la conexión con el dispositivo o bien con el archivo de registro de actividades el mismo servicio manda un registro al Log File informando acerca del error.

Este módulo fue desarrollado en el lenguaje de programación C# debido a que la aplicación dentro de la terminal portátil está desarrollada en el mismo lenguaje.

Interfaz web y comunicación con la red de medidores inteligente

Dentro de la fase de desarrollo para el módulo de interfaz web se optó por la implementación de un Servlet debido a que esta tecnología brinda la facilidad de correr del lado del servidor en una arquitectura Cliente – Servidor, minimizando de este modo el uso de recursos por parte del cliente. (Hunter Jason, 2001)

El Framework Prime Faces brindó aún más facilidades dentro del desarrollo al ofrecer controles interactivos y visualmente atractivos, listos para ser implementados y modificados según las necesidades del programador; lo cual disminuye notablemente el tiempo de desarrollo y minimiza esfuerzos.

En cuanto a la interacción del módulo de interfaz web con la red de medidores; específicamente con el dispositivo colector se plantea el desarrollo de un socket TCP IP el cual será programado también en Java por motivos de compatibilidad, abriendo un canal de comunicación permanente con el dispositivo en espera de un mensaje ya sea desde la interfaz web o desde la red de medidores.

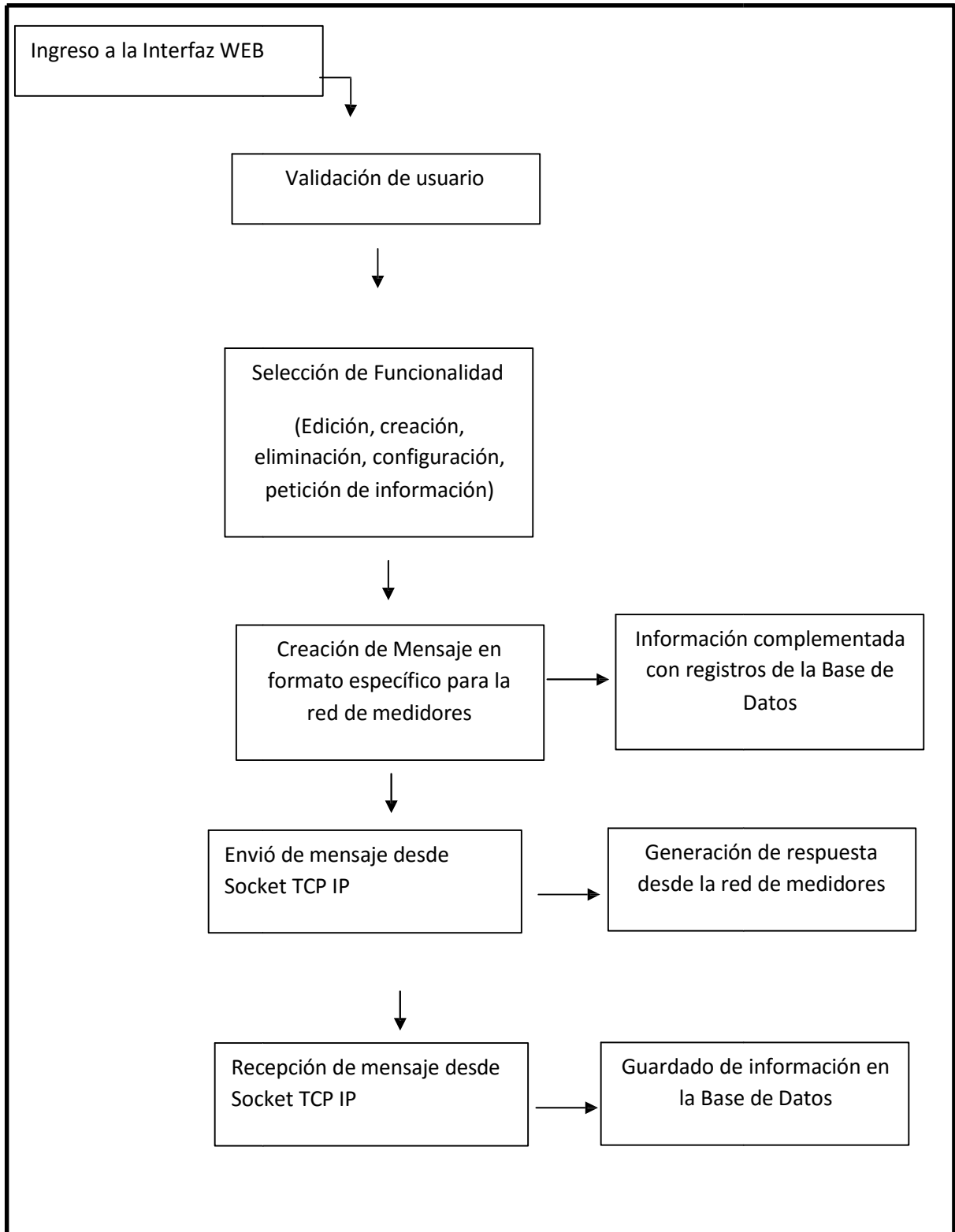


Fig. 2. Diagrama a bloques de la funcionalidad del módulo de interfaz web y socket TCP.

En la figura 2 podemos observar el proceso de flujo de datos a través del módulo de interfaz web para el Head End, el proceso comienza cuando un oficinista de CFE ingresa vía internet al Servlet accedendo a funcionalidades de alta, edición, borrado de usuarios y dispositivos que se encuentran dentro de la red de medidores inteligente. Una vez que levanta la orden con la operación, el Servlet se encarga de ajustar el mensaje complementándolo con información obtenida de la base de datos acerca de identificadores de los dispositivos como por ejemplo a que gabinete pertenece un medidor. Este mensaje es enviado a través del socket hacia el colector en específico para que este encamine el mensaje hasta su destino de manera correcta. Una vez que el mensaje es recibido, se genera una respuesta con el estatus de la ejecución el cual es reenviado desde el socket al Servlet para que este tome las medidas indicadas, ya sea guardar los registros en la base de datos; o bien, reintentar el envío del mensaje según sea programado.

Comentarios Finales

A lo largo del desarrollo del Head End, se observó una continua evolución en los requerimientos para un mejor funcionamiento e integración de tecnologías actuales, por lo cual debido a la naturaleza cambiante del sistema encontramos una serie de facilidades en las técnicas previamente mencionadas.

Las metodologías brindaron agilidad en la adaptación de requerimientos en tiempo de desarrollo, evitando retrasos o reestructuraciones de estos, cada cambio fue documentado de manera pertinente, evaluado en cuanto a factibilidad, analizado para su priorización y desarrollado según la calendarización. Del mismo modo el buen planteamiento de los requerimientos permitió incluso la colaboración en línea de los miembros del equipo cuando estos se encontraban en diferentes zonas geográficas debido a una delegación de responsabilidades clara y oportuna. La implementación de una metodología ágil, además de la modularización del sistema facilitó el cumplimiento de los requerimientos en tiempo al desfragmentar el sistema en operaciones sencillas de programar y entender para los desarrolladores ahorrando tiempos y esfuerzos.

Resumen de resultados

En el presente trabajo se presenta una combinación de metodologías para el desarrollo del software las cuales fueron integradas para facilitar el proceso de creación del sistema informático de tipo Head End.

Cada metodología o técnica de programación fue seleccionada cuidadosamente por el líder del equipo de desarrollo entregando como resultado funcionalidades confiables en tiempo y forma.

REGISTRO DE PRUEBAS	
FUNCIONALIDAD	%
Daemon para comunicación con dispositivos Hand Held	100
Conexión del Sistema con la base de datos de Oracle	100
Modelado de base de datos e integración con Hibernate en Java	100
Módulo de Acceso WEB (Servlet)	100
Funcionalidad del módulo de interfaz web	100
Pruebas de comunicación socket – Red de medidores	PENDIENTE

Tabla 1. Registro de pruebas realizadas a los módulos del sistema

Como puede observarse en la tabla número 1 se realizó un registro resumido de las principales funcionalidades del Head End desarrollado, mostrando el porcentaje de completitud alcanzado en el tiempo establecido para el desarrollo de los requerimientos; lo cual demuestra la fiabilidad de integración de las técnicas de programación y metodologías implementadas dentro de este artículo así como una buena visión y planeación por parte del equipo desarrollador, con resultados comprobados al ser presentados frente al cliente el cual aprobó las entregas.

Conclusiones

La integración de todas las metodologías mencionadas en el presente texto brinda agilidad y rapidez en todo el proceso de desarrollo de software, para el caso del presente sistema; se agilizaron los tiempos de desarrollo por cada integrante del equipo dando como resultado la programación de funcionalidades que cumplen de manera satisfactoria con los requerimientos establecidos, y aunque estos a su vez eran reestablecidos o complementados durante el tiempo de desarrollo, no mostraron mayores complicaciones gracias al enfoque tomado por el líder del equipo de desarrollo.

Debido a la naturaleza del proyecto hubo momentos en que el equipo tuvo que estar trabajando de manera remota sin presentar ninguna complicación los desarrolladores pudieron continuar con su trabajo y una vez que la funcionalidad quedaba terminada esta era agregada como un módulo más dentro de la aplicación probando una completa compatibilidad gracias a un correcto establecimiento y distribución entre los miembros del equipo de los requerimientos.



"2015. Año del Generalísimo José María Morelos y Pavón"

TECNOLÓGICO NACIONAL DE MÉXICO
Instituto Tecnológico de Apizaco



Trabajos Futuros

Como se mencionó anteriormente la fase de desarrollo para el modulo del socket TCP será presentada en trabajos posteriores.

Actualmente se está desarrollando la funcionalidad para el socket TCP y la adaptación de tablas bajo el estándar ANSI C 12.18 y 12.19 para el manejo de los mensajes hacia la red de medidores.

En conjunto con funcionalidad para graficación de uso estadístico del sistema, que se planteó como uno de los requerimientos funcionales con menor prioridad.

Referencias

Zheng, Jixuan, Gao, D.W., Lin, Li, "Smart Meters in smart Grid; An Overview", IEEE, 2013

Pries, Kinm H., Quigley, Jon M. "Scrum Project Management", CRC Press, 2010.

Goal Anita "Computer Fundamentals" Pearson Education India, 2010.

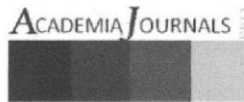
James Elliot , "Hibernate, A Developers Notebook", O' Reily Media, Inc., 2004.

Kogent Learning Solutios Inc., "C# Programming,Covers .NET3.5, Black Book, Begginers Ed.",DreamTEch Press,2010.

Hunter Jason, Crawford William, "Java Servlet Programming",O'Reilly,2001.

CICS.ACADEMIAJOURNALS.COM

Congreso Internacional de Investigación
de
Academia Journals
en Ciencias y Sustentabilidad



Certificado

otorgado a

Lic. Agustín Sánchez Atonal
M.C. José Juan Hernández Mora
M.C. Ma. Guadalupe Medina Barrera
Lic. Carlos Domínguez Galván

por su ponencia intitulada

Desarrollo de un sistema informático de tipo "Head End" para redes inteligentes

No. ponencia X070

la cual fue presentada en el Congreso que se desarrolló los días 27 al 29 de mayo del año 2015 en Tuxpan, Veracruz, México y se publicó en el portal de Internet CICS.AcademiaJournals.com con ISSN 2169-6160 Online, memorias en CDROM con ISSN 2169-6152, y libro electrónico "Investigación en las Ciencias con Pertinencia" con ISBN 978-1-939982-09-4



Dr. Edalid Álvarez Velázquez
Presidente de la Comisión Organizadora
Directora de la Facultad de Contaduría
Universidad Veracruzana Región Poza Rica-Tuxpan



Dr. Rafael Moras
Editor, AcademiaJournals.com
Profesor de Ing. Industrial y Administrativa
St. Mary's University, San Antonio, TX, EEUU

FACULTAD DE CONTADURIA



ZONA POZA RICA-TUXPAN
DIRECCION

X070