



TÍTULO

Módulo de migración de información de OCS Inventory NG 2.0.5 a
GRHS 1.0

Trabajo de Diploma para optar por el Título de Ingeniero
en Ciencias Informáticas

AUTORES

Bladimir Almeida Rodríguez

Ronal Ledesma López

TUTORES

Ing. Osbel Ledesma Peral

Ing. Jenny De la Rosa Pasteur

La Habana, 2015

“Año 57 de la Revolución”

DECLARACIÓN DE AUTORÍA

Declaramos que somos los únicos autores de este trabajo y accedemos a que el Centro de Telemática (TLM) de la Facultad 2 y la Universidad de las Ciencias Informáticas (UCI) hagan uso del mismo para su beneficio propio.

Para que así conste firmo la presente a los _____ días del mes de _____ del año _____.

Bladimir Almeida Rodríguez

(Autor)

Ronald Ledesma López

(Autor)

Ing. Osbel Ledesma Peral

(Tutor)

Ing. Jenny De la Rosa Pasteur

(Tutor)

DATOS DEL CONTACTO

TUTOR

Nombre y Apellidos: Osbel Ledesma Peral..

Sexo: Masculino.

Institución: UCI.

Dirección de la Institución: Carretera a San Antonio de los Baños, KM 2 ½, Torrens, municipio de La Lisa, Cuba.

Correo electrónico: oledesma@uci.cu.

Categoría docente: Trabajador.

Área: Centro de Telemática.

Título de la especialidad de graduado: Ingeniero en Ciencias Informáticas.

Año de graduación: 2010.

Institución donde se graduó: Universidad de las Ciencias Informáticas.

TUTOR

Nombre y Apellidos: Jenny De la Rosa Pasteur

Sexo: Femenino.

Institución: UCI.

Dirección de la institución: Carretera a San Antonio de los Baños, KM 2 ½, Torrens, municipio de La Lisa, Cuba.

Correo electrónico: jdelarosa@uci.cu

Categoría docente: Trabajador.

Área: Centro de Telemática.

Título de la especialidad de graduado: Ingeniero en Ciencias Informáticas.

Año de graduación: 2010

Institución donde se graduó: Universidad de las Ciencias Informáticas.



Si lo puedes
soñar,
lo puedes
lograr.

Walt Disney

AGRADECIMIENTOS

De Bladimir:

Quiero agradecer primero que todo a Dios por brindarme su fé en los buenos y malos momentos, y permitir que el ángel que responde al nombre de Lourdes Recio Recio vele por mí desde el cielo.

Quiero agradecer a todo los profes que tuvieron que soportarme durante estos 5 años, a Mendes, Barbara Laborí, Julio, Aliennys y otros que supieron guiarnos hasta aquí, cada uno aportando su pedacito.

Agradecer a toda mi familia la cual ha contribuido de manera especial a que yo esté aquí hoy alcanzando esta meta. En especial a mi abu, mi madre, mi abuelo y mi padre.

A todos los amigos que he hecho aquí, el pollo, Raúl, el rolo, armo, el mou, Enrique, el pabli, el chino, el pelú y otros muchos que han tenido que aguantar mis tostones.

Agradecer a mi hermano Ledesma, compañero de tesis, y un miembro más de mi familia, al Luiso por soportarme durante los 5 años como compañero de equipo, a Ada por su abnegación y desinterés.

A los tutores Osbel y Jenny, capaces de encausar a esta pareja de tesistas por el camino correcto, a mucha gente que sin dudas ha formado parte de mi vida en la universidad y puede ser que su nombre no venga a mi mente en estos momentos, pero de seguro los llevo en el corazón.

A mi novia Ania que ha mostrado los dotes de mujer intransigente, por permitirme terminar la universidad y convertirse en padre y madre al mismo tiempo.

A mi suegra Olga por preocuparse por mí.

A Yuly, a Javie por ser los mejores amigos que uno puede tener.

A mis amigos de la vida, el carly, ale, yoandy, ferna, mi primo juanka.

A mi tío Juan Carlos, mis tías Susanas y mi abuela Victoria.

A todos gracias....

AGRADECIMIENTOS

De Ronal:

Agradecerle a los profes que me impartieron clases y supieron contribuir a mi aprendizaje, a todos gracias.

A mi familia por apoyarme a cada momento, y guiarme por el camino correcto.

A mis amigos de la Universidad, me llevo una gran imagen de todos, sepan que pueden contar conmigo por siempre.

A mi primo Osbel por los buenos momentos compartidos.

A la tutora Jenny por jalarme las orejas cuando me descarriaba, y sacar el mayor provecho de mí, te aprecio, gracias.

A mi compañero de tesis y hermano Blade, por guapear conmigo en las buenas y las malas, y no dejarme dormir 5 minutos más cuando se lo pedía.

Al Yase y el Rolo, hermanos también, gracias por todas las experiencias vividas.

A Ada por todo su apoyo y ser excelente compañera.

A todos los compañeros del aula, por permitir mis repentinas incursiones en los debates de reunión y de clases.

A todos muchas gracias.....

DEDICATORIA

De Bladimir:

Quiero dedicar este hermoso logro a 5 personas fundamentales en mi vida.

A mi abuelita Lourdes por enseñarme a proponerme metas y no descansar hasta lograrlas, por malcriarme y dar su vida por mí, por haber inundado de amor su crianza hacia mí, ahora estoy aquí como ingeniero y quiero que donde quieras que estés sepas que todos los logros de mi vida tienen un nombre y ese es: "Lourdes". Te extraño mi viejita linda, estoy donde siempre quisiste, mi corazón es y será siempre tuyo.

A mi madre, que de no ser por ella no hubiera sobrepasado los obstáculos que surgieron en el camino, mami linda gracias por insistir que siempre se puede, por ser mi estimulación, mi gran amiga, mi paño de lágrimas y por lanzarte conmigo en esta carrera de la vida.

A mi bebé el cual desde que nació ha sido la motivación más importante, el cual ha estado sometido a una angustiada lejanía para obtener este título, Fer te dedico este momento glorioso, y deseo convertirme en un ejemplo a seguir para ti.

A mi abuelo Almeida, por darme todo su amor, por enseñarme a ser un hombre de bien, por estar siempre ahí para mí, por hacerme fuerte para enfrentar la partida de la vieja. Abuelito no tengo palabras para decirte todo lo que te quiero, este título es tuyo, lo sabes.

A mi papá Bladimir por ser un ejemplo a seguir tanto en el plano personal como en el profesional, por darme su cariño, su apoyo, su amistad y sus consejos. Papi quisiera ser como tú.

De Ronal:

Dedico este gran triunfo a mis padres y a mi abuela, quienes merecen mucho más.

Resumen

El centro productivo TLM perteneciente a la Facultad 2 de la UCI desarrolló la versión 1.0 del Sistema Gestor de Recursos de Hardware y Software (GRHS), encargado de realizar un inventario del hardware y software en una red de ordenadores. GRHS no permite relacionarse con el sistema Open Computer Software Next Generation (OCS Inventory NG), sistema similar que detecta anomalías en la red, debido a que carece de la funcionalidad que permita migrar los datos que son arrojados por los inventarios realizados por OCS Inventory NG y que tienen una similitud a los que genera GRHS en sus inventarios. Este trabajo se basa en la inserción de un módulo a la herramienta GRHS en aras de suplir dicha carencia. Con la culminación de este trabajo de diploma se puede mantener la disponibilidad de la información en GRHS sobre los inventarios realizados que se encuentren en la Base de Datos (BD) de OCS Inventory NG. El desarrollo de este sistema está conducido por las especificaciones de la metodología XP, obteniendo los artefactos de las distintas fases de trabajo. Se utiliza el marco de trabajo Xilema-Base-Web el cual emplea como frameworks base a Django basado en el lenguaje de programación Python, además de disponer de los frameworks de JavaScript, JQuery y Backbone para la programación del lado del usuario.

Palabras clave: datos, GRHS, inventario, migración, OCS Inventory NG

Índice

DECLARACIÓN DE AUTORÍA..... I

DATOS DEL CONTACTO..... II

AGRADECIMIENTOS III

DEDICATORIA..... IV

Resumen V

Índice de tablas..... XI

Índice de figuras XIII

Introducción..... 1

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA..... 5

1.1 Introducción..... 5

1.2 Conceptos fundamentales..... 5

1.2.1 Migración de datos..... 5

1.2.2 Factores críticos de éxitos de una migración de datos..... 6

1.2.3 Datos 7

1.2.4 Base de Datos (BD) 7

1.2.5 Modelos de dato relacional..... 7

1.3 Descripción de herramientas que intervienen en el proceso de migración 8

1.3.1 Gestor de Recursos de Hardware y Software (GRHS)..... 8

1.3.2 Open Computer System Inventory Next Generation..... 8

1.4 Estudio de herramientas internacionales que realizan el proceso de migración de SGBD 9

1.5 Estudio de herramientas nacionales que realizan el proceso de migración de SGBD 11

1.6 Metodología de desarrollo 12

1.6.1 Extreme Programming (XP) 12

1.7 Lenguajes de programación	13
1.7.1 Python 2.7.3	13
1.7.2 Hojas de Estilo en Cascada (CCS)	14
1.7.3 Lenguaje de Marcación de Hipertexto 5 (HTML 5).....	14
1.7.4 JavaScript.....	15
1.8 Sistema Gestor de Base de Datos	16
1.8.2 PostgreSQL 9.1.....	16
1.8.3 MySQL 5.5.40.....	16
1.9 Herramientas de administración de SGBD	16
1.10 Herramienta CASE (Computer Aided Software Engineering) Visual Paradigm for UML Personal Edition 8.0	17
1.11 Lenguaje Unificado de Modelado (UML)	18
1.12 Tecnología	18
1.12.1 Ajax.....	18
1.12.2 JavaScript Object Notation (JSON).....	18
1.13 Marco de trabajo Xilema-Base-Web	19
1.13.1 Django 1.4.5	19
1.13.2 Backbone 1.1.0	20
1.13.3 JQuery 1.9.0.....	20
1.14 Entorno de desarrollo integrado (IDE)	21
1.14.1 Eclipse v3.8 (Pydev 4.0)	21
1.15 Conclusiones del capítulo	21
CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA, EXPLORACIÓN, PLANIFICACIÓN Y DISEÑO	22
2.1 Introducción	22
2.2 Descripción del módulo propuesto	22
2.3 Descripción del negocio	23

2.3.1 Observaciones.....	25
2.4 Fase Exploración.....	25
2.4.1 Historias de Usuario.....	26
2.5 Características no funcionales del módulo.....	28
2.5.1 Software.....	28
2.5.2 Hardware.....	28
2.5.3 Usabilidad.....	29
2.5.4 Legales.....	29
2.5.5 Interfaz.....	29
2.5.6 Seguridad.....	29
2.6 Fase de Planificación.....	29
2.6.1 Estimación de esfuerzo.....	29
2.6.2 Plan de iteraciones.....	30
2.6.2.1 Iteración I.....	31
2.6.2.2 Iteración II.....	31
2.6.2.3 Iteración III.....	31
2.6.2.3 Iteración IV.....	31
2.6.3 Duración de las iteraciones.....	31
2.6.4 Plan de entrega.....	32
2.7 Diseño.....	33
2.7.1 Tarjetas Clase-Responsabilidad-Colaborador (CRC).....	33
2.7.2 Modelo de datos.....	35
2.7.3 Patrón de arquitectura.....	36
2.7.3.1 Arquitectura Modelo-Plantilla-Vista.....	36
2.7.4 Patrón de diseño.....	38
2.7.4.1 Patrones GRASP.....	38

2.8 Conclusiones del capítulo	39
CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBAS.....	40
3.1 Introducción.....	40
3.2 Implementación	40
3.2.1 Diagrama de despliegue	40
3.2.2 Estándar de nomenclatura y codificación	41
3.2.2.1 Comentarios	41
3.2.2.2 Declaración de variables	42
3.2.2.3 Identificadores	42
3.2.2.4 Sentencias.....	42
3.2.2.5 Importaciones	42
3.2.3 Convenciones de codificación para JavaScript	43
3.2.3.1 Nombres de fichero	43
3.2.3.2 Comentarios	43
3.2.3.3 Declaración de variables	43
3.2.3.4 Objeto global	43
3.2.3.5 Constructores	44
3.2.3.6 Métodos	44
3.2.4 Tareas de Ingeniería.....	44
3.3 Pruebas	46
3.3.1 Métodos de prueba.....	46
3.3.1.1 Técnicas de prueba de caja blanca	46
3.3.1.2 Automatización de las técnicas de prueba de caja blanca	47
3.3.2 Pruebas unitaria	47
3.3.2.1 Iteración I.....	47
3.3.2.2 Iteración II.....	48

3.3.2.3 Iteración III.....	48
3.3.2.4 Iteración IV	48
3.3.3 Pruebas de integración	49
3.3.4 Pruebas de aceptación.....	50
3.5 Conclusiones del Capítulo.....	52
Conclusiones generales	53
Recomendaciones.....	54
Referencias.....	55
Bibliografía.....	60
Anexos.....	66
Anexo I: Transformación y migración de base de datos.....	66
Anexo II: Muestras de historias de usuario.....	73
Anexo III: Muestras de Tarjetas CRC	79
Anexo IV: Muestras de tareas de Ingeniería.....	80
Anexo V: Muestra de pruebas de aceptación.....	84

Índice de tablas

Tabla 1: Mapeo de las bases de datos de OCS Inventory NG y GRHS	24
Tabla 2: HU #1 Realizar la migración de datos de OCS Inventory NG a GRHS	28
Tabla 3: Estimación de esfuerzo	30
Tabla 4: Duración de las iteraciones.....	32
Tabla 5: Plan de entrega	32
Tabla 6: Tarjeta CRC # 1 ConfigurationUpdateDeleteRestView	34
Tabla 7: Tarjeta CRC # 2 Test_Connection	34
Tabla 8: Tarjeta CRC # 3 Migration.....	34
Tabla 9: TI # 1 Migrar información obtenida de la base de datos de OCS Inventory NG ...	45
Tabla 10: TI # 5 Adicionar Conexión a MySQL	45
Tabla 11: Prueba de aceptación # 1 HU Gestionar conexiones	51
Tabla 12: Transformación y migración de la base de datos de OCS Inventory NG a GRHS	66
Tabla 13: HU #2 Gestionar conexión al SGBD MySQL	73
Tabla 14: HU # 2 Establecer conexión con el gestor de base de datos MySQL.....	75
Tabla 15: HU # 4 Mostrar información migrada de OCS Inventory NG a GRHS	76
Tabla 16: HU # 5 Editar el nombre de los host migrados	77
Tabla 17: HU # 6 Mostrar mensaje acerca de la migración de información	78
Tabla 18: Tarjeta CRC Insert_agent_inventory.....	79
Tabla 19: Tarjeta CRC Insert_Software	79
Tabla 20: Tarjeta CRC Insert_Hardware.....	79
Tabla 21: TI # 2 Obtener información de la base de datos de OCS Inventory NG	80
Tabla 22: TI # 3 Transformar información de la base de datos de OCS Inventory NG	80
Tabla 23: TI # 4 Insertar información en la BD GRHS	81
Tabla 24: TI # 6 Editar Conexión a MySQL	81
Tabla 25: TI # 7 Eliminar Conexión a MySQL	82
Tabla 26: TI # 8 Establecer conexión con el SGBD MySQL.....	82
Tabla 27: TI # 9 Mostrar información migrada de OCS Inventory NG a GRHS.....	82
Tabla 28: TI # 10 Editar el nombre de los host migrados	83
Tabla 29: TI # 11 Mostrar mensaje acerca de la migración de información	83
Tabla 30: Prueba de aceptación # 2 HU Gestionar Conexiones.....	84
Tabla 31: Prueba de aceptación # 3 HU Gestionar Conexiones.....	86

Tabla 32: Pruebas de aceptación # 4 HU Establecer conexión con el gestor de base de datos MySQL	87
Tabla 33: Prueba de aceptación # 5 HU Mostrar información migrada de OCS Inventory NG a GRHS	89
Tabla 34: Prueba de aceptación # 6 HU Mostrar información migrada de OCS Inventory NG a GRHS	90
Tabla 35: Prueba de aceptación # 7 HU Mostrar información migrada de OCS Inventory NG a GRHS	91
Tabla 36: Prueba de aceptación # 8 HU Editar el nombre de los host migrados	92

Índice de figuras

Figura 1: Propuesta de solución	23
Figura 2: Diagrama Entidad Relación OCS Inventory NG.....	35
Figura 3: Diagrama Entidad Relación GRHS	36
Figura 4: Arquitectura Modelo-Plantilla-Vista	37
Figura 5: Capa Modelo	37
Figura 6: Capa Plantilla.....	38
Figura 7: Capa Vista	38
Figura 8: Diagrama de despliegue	40
Figura 9: Resultados de las pruebas unitarias. I Iteración	48
Figura 10: Resultados de las pruebas unitarias. II Iteración	48
Figura 11: Resultados de las pruebas unitarias. III Iteración	48
Figura 12: Resultados de pruebas unitarias. IV Iteración.....	49
Figura 13: Muestra de la prueba de integración.....	50
Figura 14: Gráfico que recoge el comportamiento de las no conformidades planteadas por el usuario en las pruebas de aceptación	50

Introducción

El empleo de la informática en la sociedad de hoy, le posibilita al hombre aprovechar los recursos computacionales en la administración de las empresas, con el fin de lograr una mejor gestión y rendimiento organizacional. Por tal razón le exige a los profesionales de esta rama realizar investigaciones en materia de tecnologías de la información, proyectar, organizar y supervisar el diseño, construcción, funcionamiento y mantenimiento de sistemas informáticos, estudiar aspectos tecnológicos de software, hardware y del tratamiento automático de la información. Además demanda llevar un estricto control sobre las características de las estaciones de trabajo de una empresa, o simplemente en un grupo de trabajo, actividad que recibe el nombre de inventario.

Con el fin de eliminar tales situaciones surgen los Sistemas de Gestión de Inventario de Red (SGIR); sistemas con soporte de datos, que acumulan información sobre los activos informáticos en una red de ordenadores, encargándose de realizar controles periódicos al hardware y software, conociéndose como inventarios de red. A lo largo de los años los desarrolladores de software han ido presentando soluciones de SGIR que han ido evolucionando de acuerdo a los avances tecnológicos.

Herramientas como el Open Computer Software Inventory Next Generation, más conocido como OCS Inventory NG es uno de los SGIR desarrollados, y de los más usados debido a su eficiencia, libre distribución y asentamiento. (2). Otro sistema es el Gestor de Recursos de Hardware y Software desarrollado por el Centro TLM, perteneciente a la Facultad 2 de la UCI, el cual es un sistema informático basado en la arquitectura cliente-servidor que permite realizar inventario de hardware y software en una red de computadoras.(3)

Tanto GRHS como OCS Inventory NG operan con un gran volumen de datos, debido al valioso trabajo que realizan con estos. Para cumplir con esta tarea se valen de herramientas informáticas como los Sistemas Gestores de Base de Datos (SGBD), los cuales han alcanzado un lugar importante en las aplicaciones informáticas donde es necesario almacenar información. Su objetivo es manipular de modo sistemático y sencillo la información; servir de interfaces entre las bases de datos, usuarios y aplicaciones. Los SGBD utilizados en estos sistemas son PostgreSQL y MySQL respectivamente. Este último de licencia dual y patrocinado por la empresa privada Oracle Corporation que posee el derecho de autor de la mayor parte del código. En cuanto a PostgreSQL es considerado como un sistema estable, del alto rendimiento y de libre distribución.

En algunas de las entidades organizacionales donde es necesario almacenar información digital (datos) se ha convertido en una necesidad reemplazar a menudo los sistemas en uso que gestionan dicha información debido al auge tecnológico actual, pues cada día surgen aplicaciones con mayor efectividad en el ámbito, por lo que los datos que contiene el sistema en explotación es preciso situarlos en el nuevo sistema, para la conservación de la información histórica. Además el reemplazo de sistemas posibilita efectuar nuevos procesos de negocios y requerimientos de usuario; así como garantizar la compatibilidad con otras aplicaciones, la actualización de versiones y la reducción de costos.

Las organizaciones que hacen uso de OCS Inventory NG que decidan usar GRHS están expuestas a perder la disponibilidad de la información que se va a migrar, pues no existe forma de obtener la información registrada en OCS Inventory NG para almacenarla en GRHS. Como resultado del análisis de la situación planteada surge el siguiente **problema científico**: ¿Cómo automatizar la migración de datos de OCS Inventory 2.0.5 a GRHS 1.0?

El **objeto de estudio** comprende el proceso de migración de datos.

Para dar solución al problema científico definido anteriormente, se propone como **objetivo general**: Desarrollar un módulo para GRHS que permita la migración de datos de OCS Inventory NG 2.0.5 a GRHS 1.0.

El **campo de acción** está orientado hacia la migración de datos de OCS Inventory NG 2.0.5 a GRHS 1.0.

De acuerdo a lo antes expuesto se expone la siguiente **idea a defender**: El desarrollo del módulo de migración de información de OCS Inventory NG 2.0.5 a GRHS 1.0 favorece la conservación de la información contenida en la base de datos de OCS Inventory NG 2.0.5 por parte de las empresas que se auxilien de este sistema y deseen reemplazarlo por el sistema GRHS 1.0.

Para desarrollar satisfactoriamente la investigación se definen un conjunto de **tareas de la investigación** que permiten darle solución al objetivo propuesto:

- Análisis de las herramientas que realicen el proceso de migración de datos desde el SGBD MySQL hacia el SGBD PostgreSQL para determinar si estas herramientas pueden ser de utilidad en el escenario de desarrollo de este módulo.

- Mapeo de la estructura de las tablas y datos de las BD de OCS Inventory y GRHS para la correcta interpretación y evaluación de los datos.
- Caracterización sobre las tecnologías, herramientas y metodologías que intervienen en el desarrollo del módulo para la migración de datos entre MySQL y PostgreSQL.
- Evaluación de los patrones de diseño existentes para determinar cuál de ellos es utilizado en la implementación del módulo de migración.
- Investigación acerca de los estándares de codificación, para precisar cuál es el establecido en este módulo de migración.
- Caracterización de los niveles de pruebas, los tipos de prueba y los métodos de prueba necesarios para validar que el módulo de migración responde a las necesidades del usuario final.

Durante la investigación y desarrollo de este trabajo de diploma se emplean distintos métodos científicos para la obtención, procesamiento y toma de decisiones, los cuales se amplían a continuación:

Métodos Teóricos:

Analítico – Sintético: Se utiliza en el proceso de análisis y revisión de artículos y documentos relacionados con el tema de migración de datos, de donde se intenta extraer los elementos esenciales que contribuyen a profundizar más sobre la temática, y acoplar de manera íntegra las distintas ideas generadas.

Histórico-Lógico: Se evidencia en la necesaria profundización teórica acerca de los estudios relacionados con el proceso de migración de datos.

Modelación: Se demuestra su uso en la modelación de diagramas que aunque no se correspondan con la metodología en uso, sirven de apoyo en determinados momentos del desarrollo.

Métodos Empíricos:

Observación: Encaminada a comprobar las regularidades que se aprecian en el proceso de migración.

Este trabajo está estructurado por 3 capítulos fundamentales, que incluye desde la fundamentación teórica hasta las pruebas realizadas al producto. A continuación se muestra la estructuración y descripción de los capítulos:

Capítulo 1. Se expone la **Fundamentación Teórica** donde se realiza el estado del arte en el cual se aborda el estudio de sistemas ya existentes que realicen la función de migración de bases de datos, también se tratan los principales conceptos que se atienden durante la investigación. Incluye además una descripción de las principales herramientas y tecnologías a utilizar en el desarrollo de la investigación, y un pormenorizado estudio sobre la metodología de desarrollo que guía el proceso de creación del software.

Capítulo 2. Se presentan las **Características del módulo** y las fases de **Exploración, Planificación y Diseño** por lo que se describe la herramienta a desarrollar, mediante la exposición de sus características principales. Además se muestran los artefactos y elementos que apoyan las fases de Exploración, Planificación y Diseño por la que transita el sistema, lo que viabiliza un mejor entendimiento de las necesidades que sostiene la herramienta a desarrollar.

Capítulo 3. Se aborda lo concerniente a las fases, **Implementación y Pruebas** del módulo de migración de bases de datos entre los gestores MySQL y PostgreSQL que corresponden a los sistemas OCS Inventory NG y GRHS. Se muestran las tareas de ingeniería correspondientes a cada HU. Se estudian los estándares de codificación para determinar cuál aplicar en la implementación del módulo. Se efectúan las pruebas precisadas por la metodología utilizada para demostrar que el módulo de migración cumpla con las expectativas generadas en el cliente. Además se aplican las pruebas de integración para verificar la funcionalidad del módulo dentro del sistema al que pertenece.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.1 Introducción

Para la creación de un sistema informático es de vital importancia que el equipo de desarrollo realice un estudio profundo de las herramientas y tecnologías más propicias a usarse de acuerdo a las tendencias actuales, esto contribuye a que no se utilicen tecnologías y herramientas obsoletas, ni la metodología inadecuada, lo que puede perjudicar la calidad del proceso productivo y con esto afectar el curso final del sistema en desarrollo. Además se considera esencial conocer acerca de los conceptos fundamentales que se manejan en el transcurso de la investigación. Este capítulo realiza el estudio de todas las herramientas y tecnologías que intervienen en el desarrollo del módulo de migración de información, los conceptos fundamentales relacionados con el proceso que se automatiza y la metodología que guía el proceso creativo.

1.2 Conceptos fundamentales

1.2.1 Migración de datos

Una migración de BD es un proceso que se realiza para mover o trasladar los datos almacenados de un formato de datos a otro, para lo cual es indispensable que antes de empezar cualquier proceso de esta naturaleza, se tenga clara y documentada la razón por la cual se está migrando, además de elaborarse la planeación detallada de las actividades contempladas. (4)

Existen diversos motivos para hacer una migración, tales como: mejorar el desempeño de la base de datos, cumplir con nuevos requerimientos de usuario, de la aplicación o políticas de seguridad, la compatibilidad con otras aplicaciones, la actualización de versiones, la estandarización de la tecnología de información en la organización, facilitar el intercambio de datos entre procesos, la reducción de costos que se puede tener al cambiar por software libre, el aumento en el volumen de datos, nuevos procesos de negocio, mejoras en la seguridad o el control de la información.(4)

Realizar la migración de base de datos conlleva a labores de transformación-adequación de los datos para que sean comprendidos por el nuevo entorno que los acoge. Es por eso que en muchas ocasiones se acude a mecanismo de intermediación para llevar la información del origen al destino. Esto permite mantener la coherencia entre los sistemas implicados en el proceso de migración. (5)

Según Susana Laura Corona (Dirección General de Servicios Cómputos, Universidad Nacional Autónoma de México), el proceso de migración cuenta con una serie de fases o etapas que no se pueden violar, los cuales son conocidos como factores críticos de éxitos.

1.2.2 Factores críticos de éxitos de una migración de datos

Se consideran elementos o aspectos fundamentales, que se deben tener presente para lograr que la herramienta de migración que se va a desarrollar tenga los resultados satisfactorios esperados, los cuales deben asegurar altos niveles de complacencia entre los miembros de la organización donde se utilizará el software.

- **Planeación:** En esta etapa se deben establecer los objetivos, alcance, estrategias, fases a seguir y la identificación de los requerimientos. Además se conforma el equipo de desarrollo y se definen los roles, asignando las responsabilidades correspondientes.
- **Mapeo de la Información:** En este período se realiza un estudio de ambos modelos de datos, tanto del actual como del nuevo, para de esta manera determinar las características de cada uno de los elementos y tablas que serán manejados y la relación entre estos. Además se analiza la correspondencia entre los campos equivalentes de ambos modelos. En esta fase es importante el intercambio con el usuario, pues en caso de presentarse determinada situación que dificulte el proceso de mostrar los datos migrados, se acuerda con el cliente la decisión a tomar.
- **Selección de la Herramienta:** Se realiza un análisis de las distintas opciones existentes escogiendo las más convenientes de acuerdo a costos, beneficios, e intereses de la organización para la cual se desarrolla la herramienta.
- **Pruebas:** Se efectúan pruebas a las aplicaciones que harán uso de la base de datos a migrar para verificar si actúan correctamente. Además se examina el proceso de migración para corroborar su completo funcionamiento. También es recomendable cerciorarse de que las consultas que le son realizadas a la base de datos a migrar pueden seguir ejecutándose normalmente, pues esta información cobra mayor importancia desde que comienza a ser provechosa para el usuario. En esta fase se estimula la optimización tanto de recursos como de tiempo
- **Migración:** Se ejecutan las actividades de extracción, transformación y carga, las cuales permiten obtener los datos desde su origen, modificarlos para cumplir con la integridad, la consistencia y las reglas del negocio e insertarlos finalmente en la BD destino. Antes de llevarlo

a cabo se revisan aspectos tales como la seguridad de la base de datos a migrar, como respaldo a la ocurrencia de un percance que afecte el proceso de migración.

- Resultados: Medición y análisis de los resultados. (4)

1.2.3 Datos

En el campo de la informática los datos son considerados una representación simbólica (numérica, alfabética o algorítmica) de un atributo o variable cuantitativa, describiendo de esa manera sucesos y entidades. Los datos simbolizan la información que el informático manipula para la creación de una solución informática. (6)

Un dato no tiene valor semántico en sí mismo, pero si recibe un tratamiento apropiado se puede apreciar la información contenida en dichos datos, la cual después resulta de gran ayuda para la toma de decisiones en el ámbito empresarial. Es de empleo común en el ámbito informático y, en general, prácticamente en cualquier disciplina científica. En programación, es la expresión general que describe las características de las entidades sobre las cuáles opera un algoritmo. (6)

1.2.4 Base de Datos (BD)

Es una colección de datos integrados, con redundancia controlada y con una estructura que refleje las interrelaciones y restricciones existentes en el mundo real; los datos que han de ser compartidos por diferentes usuarios y aplicaciones, deben mantenerse independientes de éstas, y su definición y descripción, únicas para cada tipo de dato, han de estar almacenadas junto con los mismos. Los procedimientos de actualización y recuperación, comunes, y bien determinados, habrán de ser capaces de conservar la integridad, seguridad y confidencialidad del conjunto de datos. (7)

1.2.5 Modelos de dato relacional

El modelo relacional constituye una alternativa para la organización y representación de la información que se pretende almacenar en una base de datos. Se trata de un modelo teórico matemático que, además de proporcionarnos los elementos básicos de modelado (las relaciones), incluye un conjunto de operadores (definidos en forma de un álgebra relacional) para su manipulación, sin ambigüedad posible. (8)

El carácter formal del modelo relacional hace relativamente sencilla su representación gestión por medio de herramientas informáticas. Constituye una referencia para la construcción de la gran mayoría de los sistemas de gestión de bases de datos comerciales disponibles en el mercado.

En este modelo la información se representa en forma de tablas o relaciones, donde cada fila de la tabla se interpreta como un conjunto de valores relacionados entre sí. (8)

1.3 Descripción de herramientas que intervienen en el proceso de migración

1.3.1 Gestor de Recursos de Hardware y Software (GRHS)

GRHS es desarrollado por el Centro TLM, perteneciente a la Facultad 2 de la UCI. Su funcionamiento consiste en un cliente instalado en las PC de la entidad que realiza el inventario de hardware y software y envían la información a un servidor que almacena y procesa dicha información, notificando las acciones a realizar en caso de que se detecten anomalías en los inventarios. A esta información se accede a través de una aplicación web que permite la visualización de los datos, gestión de usuarios y roles; además de la realización de reportes e informes. GRHS está integrado por 3 módulos que sostienen su funcionamiento, ellos son:

Gclient: Es el encargado de recolectar inventarios, detectar incidencias para luego enviarlo al gserver, a donde envía también las acciones ejecutadas ante determinada anomalía.

Gserver: Mediante este módulo se procesa la información recibida por Gclient. Además se responsabiliza de hacer llegar a gclient las acciones consignadas por los usuarios a través del Gadmin.

Gadmin: Se le encomienda la tarea de mostrar a los usuarios toda la información que ha sido gestionada a través del módulo gserver.

1.3.2 Open Computer System Inventory Next Generation

Es un SGIR de software libre, su licencia pertenece a GNU General Public License (GNU GPL) en su versión 2.0. Permite la realización de inventarios de hardware y software. OCS Inventory NG usa un agente que ejecuta el inventario en las computadoras clientes y un servidor de dirección que centraliza los resultados del inventario, lo que posibilita ver los resultados del inventario y dispositivos de la red descubiertos. Se hacen comunicaciones entre agentes y servidor de dirección usando los protocolos de HTTP/HTTPS. Todos los datos se estructuran en Zlib comprimido XML para reducir el promedio de tráfico de red. Esta aplicación es multiplataforma y posee funcionalidades para la realización de reportes sobre los inventarios

siendo capaz de enviarlos a través del correo electrónico. Utiliza varias tecnologías como: PHP¹, Perl y MySQL. Esta herramienta no tiene implementaciones para la detección de cambios en los inventarios, ni para la emisión de alarmas dada la ocurrencia de estos cambios. (2)

1.4 Estudio de herramientas internacionales que realizan el proceso de migración de SGBD

ESF Database Migration Toolkit Professional Edition

Es un programa que permite convertir entre diferentes formatos de base de datos. Puede conectarse directamente a MySQL, SQL Server, PostgreSQL, Oracle, Access, Excel, Paradox, Lotus, dBase, Visual FoxPro, texto y otros formatos y convertir entre estos formatos (SQL Server a MySQL, MySQL a Acces, entre otros). Permite conectarse a la base de datos origen, seleccionar las tablas y a continuación, convertir y copiar al destino. En el proceso se puede incluir todos los datos o simplemente copiar la estructura de la base de datos. También ofrece todas las opciones de conversión necesarias teniendo en cuenta las peculiaridades de los dos formatos de entrada y salida de la base de datos, además de incluir compatibilidad para las claves primarias de las tablas, índices e identificación automática. Es compatible con todos los caracteres Unicode (UTF-8, LATIN, CP1250, ASCII etc.). Esta herramienta al migrar la estructura de la base de datos no mantiene los campos que se encuentran con letras mayúsculas, estos los cambia a minúscula. Por otro lado al migrar los campos de la base de datos les pone una T como primera letra. Además de ser una herramienta que no está basada en software libre. (9)

DBConvert for MySQL & PostgreSQL

Es una herramienta de migración de base de datos que permite convertir de MySQL a PostgreSQL, MySQL a PostgreSQL Dump, MySQL a MySQL Dump, PostgreSQL a MySQL, entre otros. Opera con una base de datos completa o solo con las tablas necesarias, campos, índices y llaves foráneas. Esta herramienta al migrar los datos en la versión de evaluación no mantiene la integridad referencial entre las tablas. Por otro lado solamente migran 50 registros por tabla, sin embargo permite migrar la estructura de la base de datos manteniendo la integridad referencial entre las tablas. Esta herramienta no está basada en software libre. (10)

¹ Hypertext Pre-processor

Mysql2pgsql.perl

Es un script basado en el lenguaje Perl usado para convertir de un dump MySQL a un formato compatible de PostgreSQL. Tiene la desventaja que no migra la estructura de un atributo con comentarios, además de que los atributos no deben tener caracteres extraños, como por ejemplo la ñ. (11). La herramienta presenta algunas dificultades en el proceso de migración, tales como:

- Convierte el IF a "if" como si fuera el nombre de una tabla.
- Elimina las comillas "" de los nombres de las tablas, lo que provoca que los nombres de los campos se migren en minúsculas.
- En las sentencias INSERT al nombre de la tabla le pone una comilla doble y una simple.

my2pg.perl

Es un script escrito en Perl, con licencia gratis enfocado a la conversión de formato MySQL en PostgreSQL. La misma al generar el archivo .sql presenta errores en la migración como son:

- Coloca los comentarios a continuación del campo, lo cual no es una sintaxis válida en PostgreSQL.
- Los nombres de las tablas y los campos los genera con comillas simples, lo que no es permisible en PostgreSQL.
- En la declaración que hace del CONSTRAINTS la palabra que precede al CONSTRAINTS debe estar sin comillas.
- En las sentencias ADD CONSTRAINT agrega ON DELETE CASCADE ON UPDATE CASCADE, las cuales deben ir en los campos que lo lleven. (12)

MySQL-to-PostgreSQL 1.0

Es una herramienta para la migración entre los gestores de MySQL a PostgreSQL. Dentro de sus principales características se pueden mencionar que soporta todas las versiones de PostgreSQL, rapidez en la copia de datos, convierte los índices de todos los atributos, soporta múltiples conjuntos de caracteres, entre otras. (13). Sin embargo presenta algunas limitaciones como son:

- No migra las vistas, procedimientos almacenados y disparadores.
- La versión de prueba solo migra 5 filas por tabla.

- A todas ellas se suma el hecho de que es una herramienta que necesita ser pagada para poder hacer uso de todas sus funcionalidades.

Navicat Premium 9.0

Es una herramienta que une todos los miembros de la familia Navicat para administración de base de datos y desarrolladores. Permite que los usuarios se conecten a cuatro de las bases de datos más comunes: MySQL, Oracle, PostgreSQL y SQLite desde una única aplicación. Entre una de sus funcionalidades se encuentra la transferencia de datos entre los diferentes gestores a los que se conecta, permitiendo migrar bases de datos entre ellos. (14). Sin embargo presenta algunas limitaciones entre las que destacan:

- No migra los índices.
- Es una herramienta propietaria.

1.5 Estudio de herramientas nacionales que realizan el proceso de migración de SGBD

MigDB Tool 2.0

Herramienta de migración desarrollada en el Centro de Tecnología de Gestión de Datos (DATEC) perteneciente a la Facultad 6 de la UCI que permite la migración de los SGBD MySQL y SQLITE a PostgreSQL, facilitando la personalización del proceso de migración, permitiendo al usuario seleccionar las funcionalidades que se ajusten a las necesidades del negocio. Además posibilita visualizar el progreso del proceso de migración mediante trazas. Esta aplicación aprueba la migración íntegra de la estructura de la base de datos, datos, llaves primarias y foráneas, índices y vistas de los SGBD MySQL y SQLITE a PostgreSQL. Es una herramienta basada en software libre, no migra objetos de la base de datos como disparadores o procedimientos almacenados. (15). No es conveniente usarla para darle solución a esta migración desde OCS Inventory a GRHS ya que sus respectivas bases de datos tienen una estructura distinta, y es necesario realizar adecuaciones en ellas para facilitar la migración.

Durante el estudio que se realiza a las herramientas que efectúan el proceso de migración de base de datos entre los SGBD PostgreSQL y MySQL se determina no utilizar ninguna de estas herramientas para apoyar la investigación ya que no son convenientes. Esto se debe a que presentan características tales como:

- Migran una estructura completa de la base de datos

- Poseen licencia privativa.
- Exhiben dificultades con funciones y sentencias que no son reconocidas en PostgreSQL.

De manera general se establece que ninguna de ellas realiza el proceso de migración que necesita GRHS, lo cual se demuestra en este análisis.

A raíz de lo que se estipula anteriormente se dispone a implementar un módulo que cubra todos los pasos de la migración, capaz de satisfacer los requerimientos del negocio exigidos por el cliente, para poder lograr un módulo eficiente y que responda a las necesidades de las organizaciones.

1.6 Metodología de desarrollo

1.6.1 Extreme Programming (XP)

Las metodologías de desarrollo son las encargadas de elaborar estrategias centradas en los clientes y equipos de desarrollo, orientadas hacia la funcionalidad y la entrega. “Su objetivo es elevar la calidad del software a través de un mayor control del proceso.” (16) Esto prescinde el riesgo y la dificultad en la realización del software además da paso a una buena relación entre el cliente y el desarrollador.

Dada la diversidad de proyectos existentes, que se pueden diferenciar de acuerdo a sus características, es coherente y preciso que existan disimiles formas de darle solución a través de las metodologías de desarrollo, las que se pueden clasificar en ligeras y robustas. A continuación se exponen particularidades acerca de XP, la metodología que conduce el curso del desarrollo de este trabajo.

XP es una metodología ágil basada fundamentalmente en la interacción constante del cliente y el equipo de desarrollo, la simplicidad en las soluciones, la resistencia al cambio y el desarrollo incremental. Por ende, es la metodología recomendada ante proyectos con requisitos variables y de alto riesgo técnico. Sus principios y prácticas son de sentido común en las metodologías de desarrollo, pero en XP son llevados al extremo. Propicia y promueve el trabajo en equipo, enfocándose en la preparación y superación de cada miembro de este, como clave del éxito. El desarrollo de soluciones mediante la metodología XP es iterativo e incremental. Además provee una impersonal propiedad del código. Orienta a la realización de las pruebas unitarias para la corrección de los errores y una refactorización del código para mantener la legibilidad del mismo.

(17)

El ciclo de vida del desarrollo consta de 6 fases, pero en este trabajo se abordan solo 4, ya que el desarrollo de este módulo no abarca las fases de mantenimiento y muerte del proyecto.

- **Exploración:** Los clientes plantean a grandes rasgos las historias de usuario que son de interés para la primera entrega del producto. Al mismo tiempo el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizan en el proyecto. Se prueba la tecnología y se exploran las posibilidades de la arquitectura del sistema construyendo un prototipo.
- **Planificación del proyecto:** En esta fase el cliente define la prioridad de las historias de usuario y el equipo de desarrollo debe estimar el tiempo y esfuerzo para cada tarea.
- **Iteraciones:** Incluye varias iteraciones sobre el sistema antes de ser entregado. Todo el trabajo de la iteración es expresado en tareas de programación. El plan de entrega está compuesto por iteraciones que deben durar de una a tres semanas.
- **Producción:** La fase de producción requiere de pruebas adicionales y revisiones de rendimiento antes de que el sistema sea trasladado al entorno del cliente. (17)

Luego de abordar detalladamente XP, se decide que esta metodología es la idónea para guiar el proceso de desarrollo, ya que muchas de sus prácticas condicionan las mejoras del producto que se crea. Además este módulo se caracteriza por ser pequeño y de pocas iteraciones, con un equipo de desarrollo de dos integrantes, y la participación del cliente cobra una importancia vital. Otra de las cuestiones acerca de la utilización de XP, es que el desarrollo guiado por esta empieza en pequeño a partir de los requerimientos básicos, pero luego se van añadiendo funcionalidades que tanto el programador como el cliente crean necesarias. Al mismo tiempo XP se adapta fácilmente al cambio, por lo que si a medida que avance el proyecto surgen nuevas expectativas o ideas estas se incorporan al proceso de desarrollo, sin afectar la finalidad de la investigación. De igual forma destacar que esta metodología contiene una abundante y esclarecida bibliografía capaz de contribuir al aprendizaje.

1.7 Lenguajes de programación

1.7.1 Python 2.7.3

Es un lenguaje que favorece un código legible. Se trata de un lenguaje interpretado o de script, con tipado dinámico, fuertemente tipado, multiplataforma y orientado a objetos. Python es un lenguaje multiplataforma que posee una sintaxis simple, clara y sencilla; el gestor de memoria, la gran cantidad de librerías disponibles y la potencia del lenguaje, entre otros, hacen que desarrollar

una aplicación en Python sea sencillo y dinámico. Tiene eficaces estructuras de datos de alto nivel y una solución de programación orientada a objetos simple, pero eficaz. (18)

Dispone de muchas funciones incorporadas en el propio lenguaje, para el tratamiento de cadenas, números, archivos. Contiene una gran biblioteca de módulos que proporcionan mucha utilidad a la hora de realizar tareas de programación web. También posee bibliotecas capaces de apoyar el desarrollo del sistema vinculados con los SGBD PostgreSQL y MySQL, es el caso de Psycopg2 y MySQLdb respectivamente. Además es el lenguaje de programación utilizado para el desarrollo del sistema GRHS.

1.7.2 Hojas de Estilo en Cascada (CCS)

CSS es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML. CSS es la mejor forma de separar los contenidos y su presentación, es imprescindible para crear páginas web complejas. Separar la definición de los contenidos y la definición de su aspecto presenta numerosas ventajas, ya que obliga a crear documentos HTML o XHTML bien definidos y con significado completo. Además, mejora la accesibilidad del documento, reduce la complejidad de su mantenimiento y permite visualizar el mismo documento en infinidad de dispositivos diferentes. (19)

Entre sus beneficios específicos están:

- Control simultáneo del estilo y formato de varias páginas web.
- Aplicación de diferentes presentaciones a distintos tipos de medios (pantalla, impresión, entre otros).
- Ofrecimiento de diversas opciones para realizar una misma tarea.
- Incrementa la accesibilidad ya que elimina los elementos de presentación de marcado que se decidan.

1.7.3 Lenguaje de Marcación de Hipertexto 5 (HTML 5)

El Lenguaje de Marcación de Hipertexto (HTML) es un lenguaje de marcado que define una estructura básica y un código para la definición de una página web, basando su filosofía en la referencia. Se crea en 1986 por Tim Berners Lee; a partir de dos herramientas que existían previamente: el concepto de Hipertexto, que permite conectar dos elementos entre sí y el Lenguaje Estándar de Marcación General (SGML) para colocar etiquetas o marcas en un texto

que indique como debe verse. HTML no presenta ningún compilador, por lo tanto algún error de sintaxis que se presente este no lo detecta y se visualiza en la forma que el navegador lo entienda. Estos documentos pueden ser mostrados por los navegadores de páginas Web en Internet, como Mozilla Firefox, Internet Explorer, Netscape Navigator, Mosaic, Opera, entre otros. Esta nueva actualización del lenguaje añade nuevos elementos y atributos los cuales proporcionan nuevas funcionalidades que están acorde con las tendencias actuales de los desarrolladores. (20)

1.7.4 JavaScript

JavaScript es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas. Una página web dinámica es aquella que incorpora efectos como aparición y desaparición de textos, animaciones que se activan al pulsar botones u otros elementos y ventanas de aviso al usuario. JavaScript es una marca registrada de la empresa Sun Microsystems. Surge por la clara necesidad de que exista un lenguaje de programación que se ejecute en el navegador del usuario, de esta forma, si el usuario no rellena correctamente un campo del formulario no tiene que esperar mucho tiempo hasta que el servidor vuelva a mostrar el formulario con los errores existentes. (21)

Es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. Soporta cuatro tipos de datos, pero no es necesario declarar el tipo de las variables, argumentos de funciones, ni valores de retornos de las funciones. (21)

Se utiliza este lenguaje de programación porque brinda las siguientes utilidades:

- Controla las ventanas del navegador y el contenido que muestran.
- Programación de páginas dinámicas simples.
- Comprueba los datos que el usuario introduce en un formulario antes de enviarlos.
- Es utilizado en navegadores como (Firefox, Opera, Safari, etc).
- Optimiza los tiempos de carga y el tráfico del servidor.
- Captura los eventos realizados por los usuarios y brinda una respuesta.
- Es utilizado por Xilema-Base-Web, el cual es el marco de trabajo base con el que trabaja el proyecto GRHS.

1.8 Sistema Gestor de Base de Datos

1.8.2 PostgreSQL 9.1

Es un SGBD objeto-relacional, bajo licencia BSD². Esta licencia al contrario de la GPL ³permite el uso del código fuente en software no libre. Es el SGBD de código abierto más avanzado del mundo y en sus últimas versiones posee muchas características que solo se podían ver en productos comerciales de alto calibre. (22)

PostgreSQL utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando. Se ejecuta en casi todos los principales sistemas operativos: Linux, Unix, BSD, Mac OS, Beos, Windows, etc. Documentación muy bien organizada, pública y libre, con comentarios de los propios usuarios. Soporte nativo para los lenguajes más populares del medio: PHP, C, C++, Perl, Python, etc. Altamente adaptable a las necesidades del cliente. (22)

1.8.3 MySQL 5.5.40

MySQL es un SGBD el cual presenta la alternativa de ser de licencia dual, capaz de almacenar una gran cantidad de datos de gran variedad y de distribuirlos para cubrir las necesidades de cualquier tipo de organización. Entre sus características más notables están su velocidad y la gran cantidad de funciones que presenta acorde con las exigencias de los desarrolladores entre las que se encuentran la integración con la mayor parte de los entornos. Su administración se basa en usuarios y privilegios. Destaca también por su portabilidad pues se ejecuta en la mayor parte de los sistemas operativos permitiendo la transferencia de datos entre estos. Sobresale por su facilidad de uso pues resulta fácil de usar y de administrar, incluso con sistemas obsoletos en los que sean difíciles los trabajos de administración son potentes y flexibles. (23)

1.9 Herramientas de administración de SGBD

1.9.1 Pgadmin 3

² Berkeley Software Distribution

³ Licencia Pública General de GNU

El Pgadmin es una interfaz gráfica de administración para las bases de datos que contiene el SGBD PostgreSQL. Esta herramienta es diseñada para contestar las necesidades de los usuarios de escribirle en SQL simples preguntas a las bases de datos complejas en vías de desarrollo. Se desarrolla por una comunidad de expertos de PostgreSQL y está disponible en más de una docena de idiomas. Es de software libre, su licencia pertenece a PostgreSQL, y es multiplataforma. (24)

1.9.2 PhpMyAdmin

El PhpMyAdmin es una herramienta de software escrita en PHP, tiene como objetivo ocuparse de la administración de MySQL para el desarrollo web. Apoya una amplia gama de funcionamientos en MySQL, los más frecuentemente usados (bases de datos, índices, las relaciones, los usuarios, los permisos, etc) pueden realizarse mediante la interfaz que le permite al usuario la interacción con el gestor. Es una herramienta multiplataforma, ya que se puede utilizar en todas las distribuciones de Linux y también en Windows usándose conjuntamente con el servidor XAMPP. (25)

Para el desarrollo de este trabajo de diploma se recibe el auxilio de PhpMyAdmin para visualizar gráficamente y administrar la base de datos del sistema OCS Inventory NG, que se encuentra en el SGBD MySQL, esto permite conocer la cantidad de tablas que genera el inventario en OCS Inventory NG, además de la estructura de esta base datos, lo cual es primordial conocer, para tener un exitoso proceso de migración.

1.10 Herramienta CASE (Computer Aided Software Engineering) Visual Paradigm for UML Personal Edition 8.0

El Visual Paradigm for UML Personal Edition es una plataforma de modelado que se diseñó para apoyar el trabajo de los arquitectos y diseñadores de distintos sistemas en construcción, durante todo el ciclo de vida de un software: análisis y diseño orientado a objetos, pruebas y despliegue. Favorece a la prontitud en la construcción de un software, reduciendo el costo en los que incurre el desarrollo del mismo, para que sea certificado como uno de alta calidad. (26)

Visual Paradigm es una herramienta que posee variadas y potentes características, pues se acopla a diversos tipos de herramienta. En este trabajo de diploma se decide utilizar la herramienta en la representación del modelo entidad-relación, el cual permite explicar las entidades relevantes de un sistema así como sus atributos e interrelaciones. Al mismo tiempo se emplea en la realización del diagrama de despliegue.

1.11 Lenguaje Unificado de Modelado (UML)

UML es una consolidación de muchas de las notaciones y conceptos más usados orientados a objetos, creado por Grade Booch, James Rumbaugh, e Ivar Jacobson. Se ha convertido en el lenguaje de modelado de sistemas más utilizado en la actualidad. El UML prescribe un conjunto de notaciones y diagramas estándar para modelar sistemas orientados a objetos y describe la semántica esencial de lo que estos diagramas y símbolos significan. Además incluye aspectos conceptuales tales como funciones del sistema y aspectos concretos como esquemas de base de datos y componentes reutilizables. Los mecanismos de extensibilidad incorporados le permiten a UML ser una especie de especificación que puede cubrir aspectos como el modelado de negocio. (27). Se utiliza esta herramienta en la realización del diagrama de despliegue para modelar la arquitectura en tiempo de ejecución de un sistema.

1.12 Tecnología

1.12.1 Ajax

Es un acrónimo de JavaScript asíncrono + XML. Ajax agrupa varias tecnologías independientes que se unen en novedosas y sorprendentes formas. Su uso permite mejorar completamente la interacción del usuario con la aplicación, evitando las recargas constantes de la página mediante la creación de un elemento intermedio entre el usuario y el servidor, ya que el intercambio de información con el servidor se produce en segundo plano. (28) Se evidencia en todas las peticiones que realiza el cliente al servidor.

1.12.2 JavaScript Object Notation (JSON)

Notación de objetos de JavaScript es un formato ligero para el intercambio de datos estructurados. Está basado en un subconjunto del lenguaje de programación JavaScript. Es un formato de texto que es completamente independiente del lenguaje pero utiliza convenciones que son ampliamente conocidos por los programadores de lenguajes como C, C++, C#, Java, JavaScript, Perl, Python y muchos otros. JSON está constituido por dos estructuras. Una de ella es conocida como un objeto, registro, estructura, diccionario, tabla, hash, lista de claves o un arreglo asociativo y la otra una lista ordenada de valores, en la mayoría esto se implementa como arreglos, vectores, listas o secuencias. (29). Se evidencia su uso en la transformación de los datos para que sean mostrados al usuario.

1.13 Marco de trabajo Xilema-Base-Web

Xilema-Base-Web es un marco de trabajo desarrollado en el Centro de TLM que está constituido por Django como framework base, librerías de javascript como son JQuery y Backbone, además cuenta con las pautas de diseño de la Universidad. (30)

1.13.1 Django 1.4.5

La presente investigación dispone de Django como framework web implementado sobre el lenguaje de programación Python, pertenece a la licencia BSD. Django brinda estructura al código fuente, fomentando las buenas prácticas de desarrollo web, lo que promueve un código legible y fácil de mantener. La implementación del patrón de diseño Model Template View (MTV) es una característica propia que contiene el frameworks, la cual contribuye a la organización de las distintas partes de la aplicación, y a modificar estas sin afectar cualquier otra pieza del software. Su alta escalabilidad le posibilita manejar el crecimiento continuo de trabajo de manera fluida sin perder calidad en los servicio. (31)

Entre otras de sus peculiaridades se encuentran:

- **Diseño de URL a gusto del desarrollador:** Django te permite crear URLs que sean del agrado del diseñador. Para esto se basa en el módulo de Python URLconf el cual le indica a la URL pertinente el código al que debe hacer referencia.
- **Sistema de plantillas:** Ostenta un extensivo sistema de plantillas pues cada vista HTML es codificada en Python, para hacer más factible la separación del diseño de la página de la lógica de Python.
- **Mapeador Objeto-Relacional (ORM):** Defines lo modelos de datos en Python, y luego se utiliza la API para acceder de manera dinámica a la base de datos.

Luego del exhaustivo análisis realizado se determina que el desarrollo de la herramienta se auxilie en Django, ya que este es el frameworks base utilizado por Xilema-Base-Web, el cual es el marco de trabajo que utiliza el proyecto GRHS al que pertenece el módulo en desarrollo. Además el desarrollo del sistema se apoya en el ORM y los métodos que este emplea para acceder a la numerosa información que se debe ocupar y que está contenida en la base de datos, otro factor que favorece al equipo en la utilización del frameworks es que este puede conectarse paralelamente a más de un SGBD, lo cual es una necesidad para el desarrollo de la herramienta.

1.13.2 Backbone 1.1.0

Es una librería de desarrollo perteneciente al lenguaje de programación JavaScript fundamentada en el patrón arquitectónico Modelo-Vista-Controlador. Proporciona una estructura a las aplicaciones web ya que puede representar sus datos como modelos con la unión clave-valor y ofrece también eventos personalizados. Abarca colecciones con una API⁴ de funciones innumerables, vistas con eventos de manipulación declarativa y lo conecta todo a su API existente sobre una interfaz JSON REST. Está bajo la licencia MIT⁵. (32). En este trabajo se evidencia el uso de esta herramienta en la creación de formularios, y tablas donde se muestra la información que se migra.

1.13.3 JQuery 1.9.0

Es un marco de trabajo JavaScript considerado uno de los complementos más esenciales para el desarrollo web, ya que facilita el desarrollo de aplicaciones enriquecidas del lado del cliente compatible con todos los navegadores. A veces se puede hacer referencia a JQuery como framework o incluso como un API de funciones, útiles en la mayoría de proyectos web. Su licencia pertenece a MIT y a GNU versión 2 la cual permite el uso en cualquier tipo de plataforma. Con esta librería se obtiene ayuda en la creación de interfaces de usuario, efectos dinámicos, aplicaciones que hacen uso de Ajax, etc. Con su utilización se tiene a disposición una interfaz para programar que permitirá hacer cosas con el navegador, de las cuales se tiene seguridad que van a funcionar para los visitantes a determinado sitio. Además es un producto serio, estable, bien documentado y con un gran equipo de desarrolladores a cargo de la mejora y actualización del framework. Otra cuestión interesante es la dilatada comunidad de creadores de plugins o componentes, lo que hace fácil encontrar soluciones ya creadas para implementar asuntos como interfaces de usuario, galerías, votaciones y efectos diversos. (33). En este trabajo se utiliza para controlar todas las operaciones de las acciones del usuario.

⁴ Interfaz de programación de aplicaciones

⁵ Instituto Tecnológico de Massachusetts

1.14 Entorno de desarrollo integrado (IDE)

1.14.1 Eclipse v3.8 (Pydev 4.0)

Eclipse Marketplace es una plataforma que contiene distintos plugins capaces de soportar determinados lenguajes de programación. En este caso el que se maneja en este trabajo es Pydev el cual es un plugins que permite emplear Eclipse como IDE para desarrollar en Python. Es multiplataforma y de código abierto. Utiliza avanzadas técnicas de inferencias, proporcionando características como el completamiento de código, análisis de código, refactorización, etc. El manejo de este IDE acompañado del plugins mencionado favorece al desarrollo del módulo Migración debido a la fácil integración que tienen estos con el framework Django. Además posee la integración de pruebas unitarias, consola interactiva y demás características que lo forjan como la herramienta a utilizar para el desarrollo.

1.15 Conclusiones del capítulo

En este capítulo se lleva a cabo el análisis de los principales aspectos y conceptos relacionados con la migración de datos, dejando sentadas las bases teóricas del trabajo a desarrollar. Se realiza el estado del arte de las herramientas que efectúan procesos de migración donde estén involucrados los SGBD PostgreSQL y MySQL, determinando que las herramientas existentes no cubren las necesidades del negocio, definidas por el cliente. Se fundamenta acerca de las tecnologías y la metodología más apropiadas para desarrollar el sistema.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA, EXPLORACIÓN, PLANIFICACIÓN Y DISEÑO

2.1 Introducción

Para desarrollar un sistema informático resulta vital entender la lógica de negocio que plantea el cliente, para luego realizar todas las acciones que en su final le brindarán a este la mejor solución posible. Por lo antes explicado en este capítulo se estudia la propuesta del negocio presentada por el cliente, y como respuesta se plasma el mapeo de las bases de datos de ambos sistemas tanto OCS Inventory NG y GRHS, para entender las particularidades de ambas bases de datos y hacia donde se debe guiar el proceso de migración. Además se describe la herramienta a desarrollar, por lo que se exponen sus características principales. Al mismo tiempo se muestran los artefactos que apoyan la fase de exploración por la que transita el sistema, lo que viabiliza un mejor entendimiento de las necesidades que sostiene la herramienta a desarrollar. También se lleva a cabo la fase de Planificación que se encarga a través de los artefactos que establece esta metodología realizar estimaciones lo más acertadas posibles, para que el cliente reciba un producto de calidad y en el tiempo adecuado. Por último se expone el diseño del módulo mediante las Tarjetas CRC Clase-Responsabilidad-Colaborador y los patrones que se aplican.

2.2 Descripción del módulo propuesto

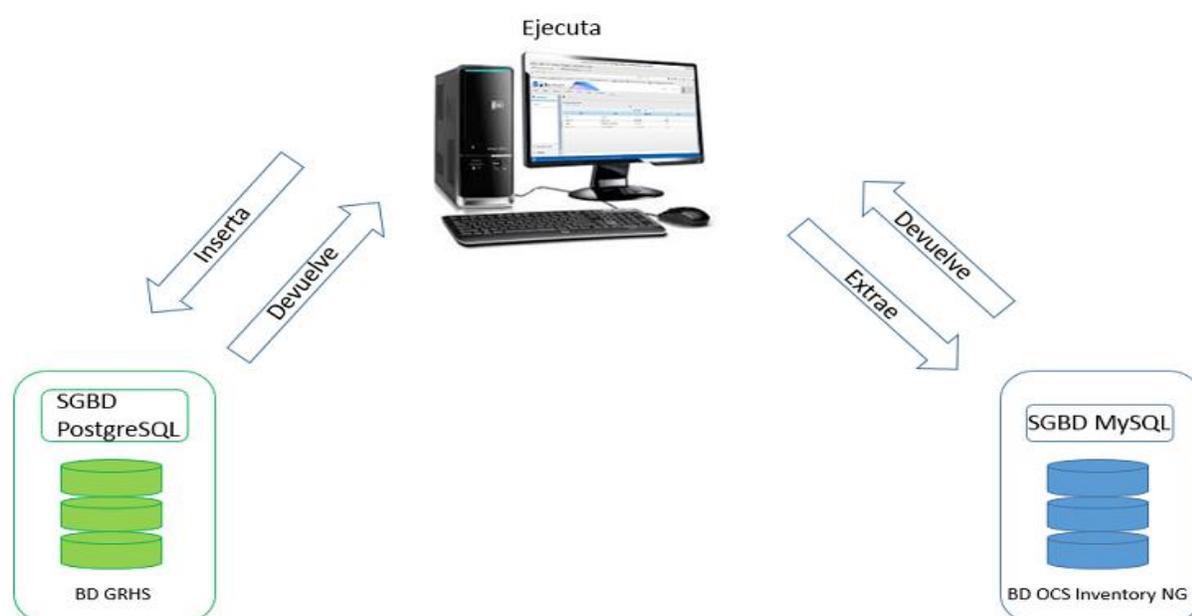
Para darle solución al problema planteado se decide realizar un módulo de migración de datos para la herramienta GRHS 1.0, en específico del lado del servidor; que permita trasladar los datos generados por OCS Inventory NG 2.0.5 a esta, teniendo como SGBD MySQL y PostgreSQL respectivamente. El módulo posibilita a las instituciones que utilizan OCS Inventory NG en su versión 2.0.5 y deseen que sus inventarios los realice GRHS 1.0 no pierdan la disponibilidad de su información. Este ofrece la opción de gestionar conexiones para el SGBD MySQL. La información se publica a través de interfaces que haciendo uso de tablas paginadas muestran la información de software y hardware correspondiente a los host migrados. Brinda la posibilidad de editar el nombre de los host migrados y adecuarlos de acuerdo a los intereses de la organización. Ofrece indicaciones a los usuarios sobre el proceso de migración, mediante un mensaje que bloquea la pantalla y no permite acceder al resto de la aplicación, hasta que no culmine dicho proceso. Además el módulo se encuentra internacionalizado sobre el sistema GRHS y puede ser integrado en cualquier aplicación que haga uso de Xilema-Base-Web, marco de trabajo que utiliza el centro Telemática. La ejecución del proceso está pre-condicionada por el siguiente procedimiento:

Capítulo 2: Características del Sistema, Exploración, Planificación y Diseño

1- Ingresar los datos para la conexión al SGBD MySQL: El usuario al seleccionar conectarse al SGBD, introduce los datos requeridos (usuario de la base de datos, nombre de la base de datos, localización y puerto) luego añade esa conexión a una lista de conexiones.

2- Ingresar contraseña del SGBD: Se accede a las conexiones existentes, y se selecciona la base de datos que se quiere migrar, se introduce la contraseña requerida y al conectarse automáticamente comienza la migración de la información contenida en esa BD hacia la BD del sistema GRHS.

Figura 1: Propuesta de solución



La imagen anterior describe el proceso de migración que se implementa, el cual consiste en extraer la información que genere OCS Inventory NG y que sea similar a la generada por GRHS durante la realización de un inventario. Luego se ejecuta la transformación de estos datos para que puedan ser asimilados por el SGBD PostgreSQL. Este último acoge esos datos, para cuando desde el sistema se consulten, sean mostrados.

2.3 Descripción del negocio

El realizar el mapeo de las BD de ambos sistemas es válido para que el equipo de trabajo se constatará de las diferencias estructurales existentes entre estas. Para demostrar lo anteriormente expuesto, se hace uso de una tabla que recoge las tablas semejantes que

Capítulo 2: Características del Sistema, Exploración, Planificación y Diseño

contienen cada BD, y el número de columnas referentes a esta, además se expone que tabla(s) de la BD del sistema GRHS recibe(n) la información de la BD del OCS Inventory NG.

Tabla 1: Mapeo de las bases de datos de OCS Inventory NG y GRHS

OCS Inventory		GRHS	
Tablas	Columnas	Tablas	Columnas
Hardware	33	Agent	3
		Computer	2
		SystemUser	4
		OperatingSystem	3
		Microprocessor	6
Controllers	8	Controller	3
Networks	14	NetworkInterface	6
softwares	10	Program	5
memories	10	RAM	7
monitors	7	Monitor	3
inputs	8	Keyboard	4
		Mouse	4
storages	10	CDROM	4
		SystemUser	4
		OperatingSystem	3

Capítulo 2: Características del Sistema, Exploración, Planificación y Diseño

		Microprocessor	6
Bios	9	Bios	3
Printers	5	Printer	3

2.3.1 Observaciones

Durante la actividad de mapeo de las bases de datos de ambos sistemas se pudo concluir que GRHS para inventariar tiene en cuenta elementos que no aprecia el OCS Inventory NG como es el caso de las motherboard de los host, es por eso que este elemento no se somete al proceso de migración. Otra desavenencia entre estos sistemas se produce cuando componentes que valora el OCS Inventory NG no guardan relación con el establecimiento de esta información en el GRHS, tal es el caso de las particiones de almacenamiento de cada computadora, y demás dispositivos de almacenamiento de entrada, ya que OCS Inventory NG no establece relación de que partición(es) pertenecen a cada dispositivo de almacenamiento. Los antivirus son inventariados por ambos pero no son tratados de la misma manera, pues el OCS Inventory NG lo evalúa como un software, mientras la herramienta del Centro de TLM le da un tratamiento especial, brindando información acerca del mismo. A través de estas observaciones se evidencia que el OCS Inventory NG no proporciona datos del BIOS para ser migrados como son Expansion, Function, Standard y Bootdevice. Como incongruencias se notan además que el OCS Inventory NG no es capaz de inventariar más de un usuario por diferentes sesiones de usuario, es decir que reemplaza el usuario de cada sesión cuando se manda la información al servidor de bases de datos.

Para hacer una explicación más exacta del proceso de transformación remítase al **Anexo I**, donde se evidencia mediante una tabla descriptiva los aspectos de OCS Inventory que son migrados y qué columnas de GRHS recibe tal información.

2.4 Fase Exploración

La fase de Exploración es el espacio que crea XP para que los clientes expongan las Historias de Usuario (HU) de mayor relevancia para el negocio, por lo que se convierte en la primera entrega del producto. A la vez el equipo de desarrollo se adapta a las tecnologías, herramientas y prácticas que se disponen en el proyecto. Además se exploran las posibilidades de la

Capítulo 2: Características del Sistema, Exploración, Planificación y Diseño

arquitectura del sistema construyendo un prototipo. Esta fase consta de pocas semanas y está en dependencia del grado de familiaridad que tengan los programadores con la tecnología.

2.4.1 Historias de Usuario

Las HU es la técnica que usa la metodología XP para la especificación de requisitos del software. Estas cobran mayor protagonismo en la fase de Exploración donde los clientes plantean las HU en lenguaje natural e intentan modelar los requisitos desde la perspectiva de los usuarios. Según Beck “las HU deben poseer la suficiente información como para determinar cuál se implementará primero, además de posibilitar la realización de estimaciones razonables sobre la duración de cada HU.” (34). El dinamismo y la flexibilidad de las que se valen las HU, son claves en el momento de enfrentar algún tipo de modificación en los requisitos.

Las HU describen las funcionalidades que debe acometer el módulo de migración de datos para que sea completamente funcional, y su tiempo de desarrollo ideal es de 1 a 3 semanas. En estas HU se considera:

Número: El número que identifica a cada HU, debe ser consecutivo.

Usuario: El usuario que acomete la acción.

Nombre de historia: El nombre de la HU.

Modificación de la HU: Informa si la HU fue sometida a algún cambio.

Iteración asignada: Iteración en la que se implementa la HU.

Programador Responsable: Programador encargado de desarrollar esa HU.

Descripción: Breve descripción de la HU.

Observaciones: Puntualizaciones acerca de esa HU.

Prototipo de Interfaz: Prototipo de Interfaz de esa HU.

La prioridad en el negocio:

- **Muy Alta:** Se les concede a las HU que son consideradas por los clientes esenciales para el desarrollo del módulo de migración.
- **Alta:** Se les concede a las HU que son consideradas por los clientes fundamentales para la ejecución de las anteriores como parte del desarrollo del módulo de migración.

Capítulo 2: Características del Sistema, Exploración, Planificación y Diseño

- **Media:** Se les concede a las HU que el cliente tiene como funcionalidades a tener en cuenta, sin que estas afecten el módulo que se está desarrollando.
- **Baja:** Se les concede a las HU que constituyen funcionalidades capaces de asistir al equipo de desarrollo y a la estructura del módulo de migración.

El riesgo en desarrollo

- **Muy Alto:** Cuando en la implementación de las HU surgen errores que lleven a la inoperatividad del sistema.
- **Alto:** Cuando en la implementación de las HU surgen errores que son significativos, pero una parte del sistema ya es funcional.
- **Medio:** Cuando en la implementación de las HU existen errores que retrasan la entrega del producto.
- **Bajo:** Cuando pueden aparecer errores que son tratados con relativa facilidad sin que traigan perjuicios para el desarrollo del proyecto. El cliente y el equipo de desarrollo trabajan en conjunto para definir como agrupar las HU para su lanzamiento.

Estimación

- Un punto de estimación equivale a una semana de programación, una semana de programación consta de 5 días y cada día tiene 8 horas de trabajo, por lo que se establece la práctica de XP de trabajar 40 horas por semana.

Luego de un exigente intercambio con el cliente, lo que garantiza el resultado satisfactorio del software, se obtiene para la presente investigación 6 HU que son implementadas en cuatro iteraciones.

HU-1 Realizar la migración de datos de OCS Inventory NG a GRHS.

HU-2 Gestionar conexiones al SGBD MySQL.

HU-3 Establecer conexión con el gestor de base de datos MySQL.

HU-4 Mostrar información migrada de OCS Inventory NG a GRHS.

HU-5 Editar el nombre de los host migrados.

HU-6 Mostrar mensaje acerca del proceso de migración.

Capítulo 2: Características del Sistema, Exploración, Planificación y Diseño

Tabla 2: HU #1 Realizar la migración de datos de OCS Inventory NG a GRHS

Historias de Usuario	
Número: 3	Usuario: Sistema
Nombre historia: Realizar la migración de datos de OCS Inventory NG a GRHS	
Modificación de Historia de usuario: Una	
Prioridad en negocio: Muy Alta	Riesgo en desarrollo: Muy Alto
Puntos estimados: 3	Iteración asignada: 1
Programador responsable: Ronal Ledesma López y Bladimir Almeida Rodríguez	
Descripción: El sistema se encarga de extraer la información contenida en la base de datos que está en el SGBD MySQL, la transforma si es necesario y la inserta en la base de datos que está en PostgreSQL	
Observaciones: La migración se realiza si la conexión fue satisfactoria.	
Prototipo de Interfaz:	

Las siguientes HU se presentan en el **Anexo II**.

2.5 Características no funcionales del módulo

Para un correcto funcionamiento del módulo de migración se deben tener en cuenta las siguientes características no funcionales.

2.5.1 Software

Sistemas Operativos: Windows 7, Windows 8, Windows 8.1 y las distribuciones de GNU/Linux, Debian 6 en adelante, Ubuntu 12.04, Centos 6 y 7 también. Controlador de Conexión: Postgre_psycomp2, MySQLdb. SGBD: MySQL v5.5.40, PostgreSQL v9.1. Lenguaje de Programación: Python v2.7.3 Marco de Trabajo: Xilema-Base-Web.

2.5.2 Hardware

Capítulo 2: Características del Sistema, Exploración, Planificación y Diseño

Los requerimientos mínimos se basan en 500 clientes registrados en Gserver. Estos son un microprocesador de 4 núcleos, la memoria RAM 2 GB y el disco duro de 10 GB.

2.5.3 Usabilidad

El módulo debe poseer facilidades para que sea usado por personal con conocimientos básicos para operar con la herramienta de migración y con los SGBD con los que opera.

2.5.4 Legales

La Universidad de las Ciencias Informáticas posee toda la propiedad intelectual del módulo elaborado.

2.5.5 Interfaz

El módulo sigue las líneas de diseño del marco de trabajo Xilema-Base-Web por el cual fueron guiadas las interfaces del sistema GRHS, para corresponder con las normas establecidas.

2.5.6 Seguridad

Confidencialidad: Solo tiene acceso a realizar la migración y a consultar la información migrada, los usuarios con los permisos requeridos para ejecutar esta actividad.

Disponibilidad: Se puede hacer uso de la aplicación siempre que esté instalado el sistema OCS Inventory NG y las tecnologías que este utiliza, además de la aplicación de GRHS, Gserver.

Integridad: El módulo garantiza que no se pierda la información mientras ocurre el proceso de migración.

2.6 Fase de Planificación

En esta fase se evidencia uno de los valores que acoge XP para el desarrollo de un sistema, como es el caso de la comunicación. Lo cual se confirma cuando el cliente establece la prioridad de cada HU, y a raíz de esto, los programadores realizan una estimación del esfuerzo necesario que conlleva desarrollar cada una de ellas. Además se toman acuerdos sobre el contenido de la primera entrega y se acuerda la confección de un cronograma en conjunto con el cliente.

2.6.1 Estimación de esfuerzo

En XP los programadores se encargan de implementar las HU y además de esto realizan las estimaciones de esfuerzo utilizando como medida el punto. Un punto equivale a una semana de

Capítulo 2: Características del Sistema, Exploración, Planificación y Diseño

programación ideal. El desarrollo de un HU se establece entre una y tres semanas. Además el equipo de desarrollo registra la velocidad del proyecto, establecida en puntos por iteración, la cual se basa en la suma de los puntos de todas las iteraciones completadas.

Tabla 3: Estimación de esfuerzo

Iteración	Historias de Usuario	Duración Total
1	1 Realizar la migración de datos de OCS Inventory NG a GRHS	3
2	1 Gestionar conexión al SGBD MySQL	2
	2 Establecer conexión con el gestor de base de datos MySQL	1
3	1 Mostrar la información migrada de OCS Inventory NG a GRHS	2
	2 Editar el nombre de los host migrados	1
4	1 Mostrar mensaje acerca del proceso de migración de la información	2
Total	6	11

2.6.2 Plan de iteraciones

El juego de la planificación es la práctica establecida por XP para el intercambio cliente-programadores donde los primeros establecen la prioridad para las HU de acuerdo al valor para el negocio que aporten cada una de ellas, y la parte técnica realiza estimados de duración en consonancia con el esfuerzo demandado por estas. Este juego se utiliza en la planificación de cada iteración y en el de entrega, pues se especifican que HU serán implementadas y en que iteración, lo que conlleva a confeccionar un plan de iteraciones. (35)

Las iteraciones en XP tienen una corta duración de pocas semanas, y su planificación tiene como objetivo el aprendizaje tanto del cliente como del programador. No se debe presionar al programador a realizar más trabajo que el estimado, ya que se pierde calidad en el software o no se cumplirán los plazos. De la misma forma el cliente tiene la obligación de manejar el ámbito de

Capítulo 2: Características del Sistema, Exploración, Planificación y Diseño

entrega del producto, para asegurarse que el sistema tenga el mayor valor de negocio posible con cada iteración. (35)

Una vez identificadas las HU del sistema y estimado el esfuerzo exigido para la realización de cada una de ellas, se procede a la ejecución de la etapa de planificación del proyecto. Para dar cumplimiento a esta fase se decide que el desarrollo del módulo abarca cuatro iteraciones, las cuales son descritas a continuación:

2.6.2.1 Iteración I

Esta iteración comprende las HU con nivel de prioridad muy alta, creando en ella la línea base para la arquitectura del sistema, las cuales para la satisfacción plena de las necesidades del cliente es inevitable su ejecución, debido al valor que le aportan al negocio. En esta iteración se obtiene la versión 0.1 donde se implementan los procesos de extracción, transformación e inserción de estos datos; los cuales forman parte del principal proceso que es la migración.

2.6.2.2 Iteración II

Esta iteración comprende las HU con nivel de prioridad alta. Se rectifican errores o inconformidades del cliente sobre la manera en que son resueltas las HU implementadas en la primera iteración. En esta iteración se obtendrá la versión 0.2 donde se automatizan la gestión de las conexiones a la BD de MySQL y también se desarrolla la funcionalidad establecer conexión.

2.6.2.3 Iteración III

En esta iteración se implementan las HU de una prioridad media en el negocio y se rectifican errores o inconformidades del cliente sobre la manera en que son resueltas las HU implementadas en la segunda iteración. Son las encargadas de mostrar la información migrada y editarla los agentes de ser necesario.

2.6.2.3 Iteración IV

En esta iteración se implementa las HU con nivel de prioridad baja, estas responden a la creación de una animación que sea responsable de orientarle al usuario que se está realizando el proceso de migración. Este espacio propicia una versión más completa y efectiva de las funcionalidades del sistema, además de contribuir a la satisfacción del cliente de acuerdo al producto en creación.

2.6.3 Duración de las iteraciones

Capítulo 2: Características del Sistema, Exploración, Planificación y Diseño

Para lograr una mayor organización del trabajo, XP expone como alternativa la creación de un plan de duración de las iteraciones. A continuación se muestra el orden en que son abordadas las HU de acuerdo a su prioridad para el negocio.

Tabla 4: Duración de las iteraciones

Iteración	Orden de la HU a implementar	Duración Total
I	<ul style="list-style-type: none">Realizar la migración de datos de OCS Inventory NG a GRHS	3
II	<ul style="list-style-type: none">Gestionar conexión al SGBD MySQLEstablecer conexión con el gestor de base de datos MySQL	3
III	<ul style="list-style-type: none">Mostrar la información migrada de OCS Inventory NG a GRHSEditar el nombre de los host migrados	3
IV	<ul style="list-style-type: none">Mostrar mensaje acerca del proceso de migración de la información	2

2.6.4 Plan de entrega

El plan de entrega se elabora con la intención de hacerle saber al cliente las posibles entregas que se realizan durante el desarrollo del módulo. Este plan permite estimar la duración total que tiene el proyecto que en este caso empieza el 16/3/2015.

Tabla 5: Plan de entrega

Módulo	Final de la Iteración I (6/4/2015)	Final de la Iteración II (27/4/2015)	Final de la Iteración III (18/5/2015)	Final de la Iteración IV (1/6/2015)
Migración de información de OCS Inventory	V0.1	V0.2	V0.3	V1.0

NG 2.0.5 a				
GRHS 1.0.				

2.7 Diseño

El diseño se realiza para crear una estructura que organiza la lógica en el sistema, lo que permite que un cambio en una parte del sistema, no siempre requiera cambios en otra parte de este. Además un buen diseño sitúa la lógica cerca de los datos con los que opera. En la estrategia de diseño, XP propone efectuar un diseño simple que fomente la comunicación entre todos los involucrados en el proceso, siendo los clientes los más favorecidos pues con este diseño se pretende que estos se mantengan informados sobre aspectos esenciales del software.

“La práctica de un diseño simple especifica que la complejidad innecesaria y el código extra deben ser removido” (35), poniéndose de manifiesto la refactorización en el momento de mejorar la legibilidad del código, para hacerlo más flexible facilitando los cambios posteriores. Según Kent Beck “en cualquier momento el diseño adecuado para el software es aquel que: supera con éxito todas las pruebas”, no tiene lógica duplicada, refleja claramente la intención de implementación de los programadores y tiene el menor número posible de clases y métodos. (35). Además Pressman asegura que el diseño es el lugar donde se fomenta la calidad del software. (36)

2.7.1 Tarjetas Clase-Responsabilidad-Colaborador (CRC)

XP para describir las clases que participan y su interacción se vale del artefacto tarjetas CRC las cuales se caracterizan por la sencillez de su uso, facilidad de su manejo y en la rapidez con la que se pueden escribir, siendo coherente todo esto con la simplicidad de XP. Estas permiten el intercambio entre las personas involucradas en el desarrollo del sistema, siendo capaces de representar determinado objeto. “La clase a la que pertenece el objeto está posicionada en la parte superior de la tabla, la responsabilidad del objeto se establece en la parte izquierda y esta debe contener información suficiente y clara acerca de su función, por último en la parte derecha se muestra la clase que colabora con cada responsabilidad.”(37) Con la realización de estas tarjetas se promueve el trabajo en grupo. Esta técnica exige la ejecución del proceso CRC, el cual a través de su etapa lluvia de ideas, genera las clases fundamentales del proceso de migración.

Capítulo 2: Características del Sistema, Exploración, Planificación y Diseño

A continuación se muestran 4 tarjetas CRC perteneciente a este trabajo, generadas para el diseño del módulo migración de datos de OCS Inventory NG 2.0.5 a GRHS 1.0. El resto de las tarjetas CRC desarrolladas se encuentran en el **Anexo III** de la investigación.

Tabla 6: Tarjeta CRC # 1 ConfigurationUpdateDeleteRestView

Tarjetas CRC	
Clase: Configuration_Migration	
Responsabilidades	Colaboraciones
<p>Es la encargada de recibir los datos que son introducidos para lograr la conexión a la base de datos.</p> <p>Es la responsable verificar que no existan errores en la prueba de conexión y que existan datos para migrar, para posteriormente desencadenar el proceso de migración.</p>	<p>Test_Connection</p> <p>All_Ready</p>

Tabla 7: Tarjeta CRC # 2 Test_Connection

Tarjetas CRC	
Clase: Test_Connection	
Responsabilidades	Colaboraciones
<p>Es la encargada de realizar una prueba de conexión a la base de datos.</p>	<p>Configuration_DATABASES</p>

Tabla 8: Tarjeta CRC # 3 Migration

Tarjetas CRC	
Clase: Migration	

Capítulo 2: Características del Sistema, Exploración, Planificación y Diseño

Responsabilidades	Colaboraciones
Es la encargada de extracción e inserción de la información referente a los componentes de hardware y software de cada computadora.	Insert_agent_inventory Inventory_dependencies Insert_Software Insert_Hardware
Acomete acciones de limpieza a los datos de hardware y software, en caso de que estos lo requieran.	Clean_Hardware Clean_Software

2.7.2 Modelo de datos

Figura 2: Diagrama Entidad Relación OCS Inventory NG

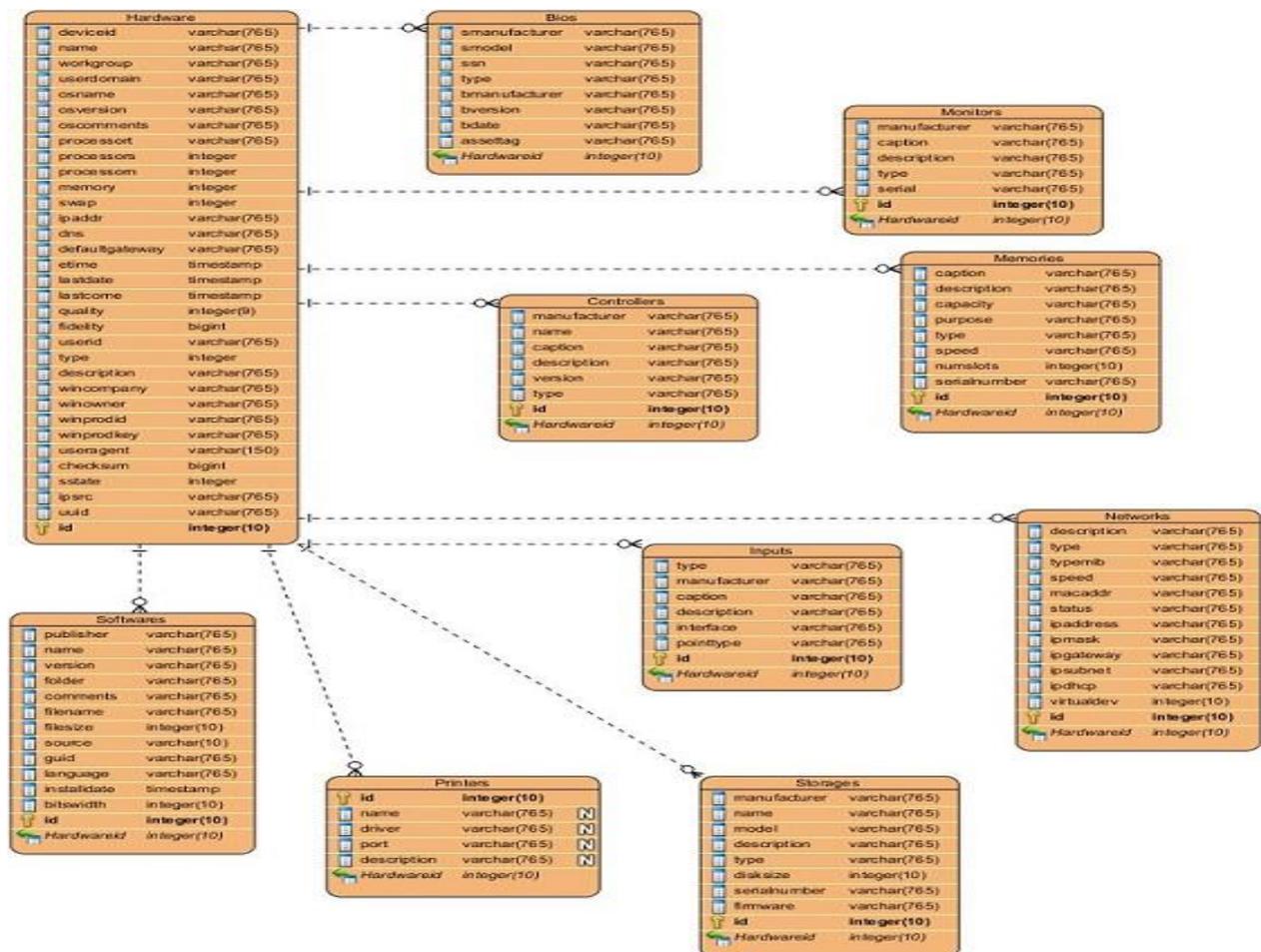
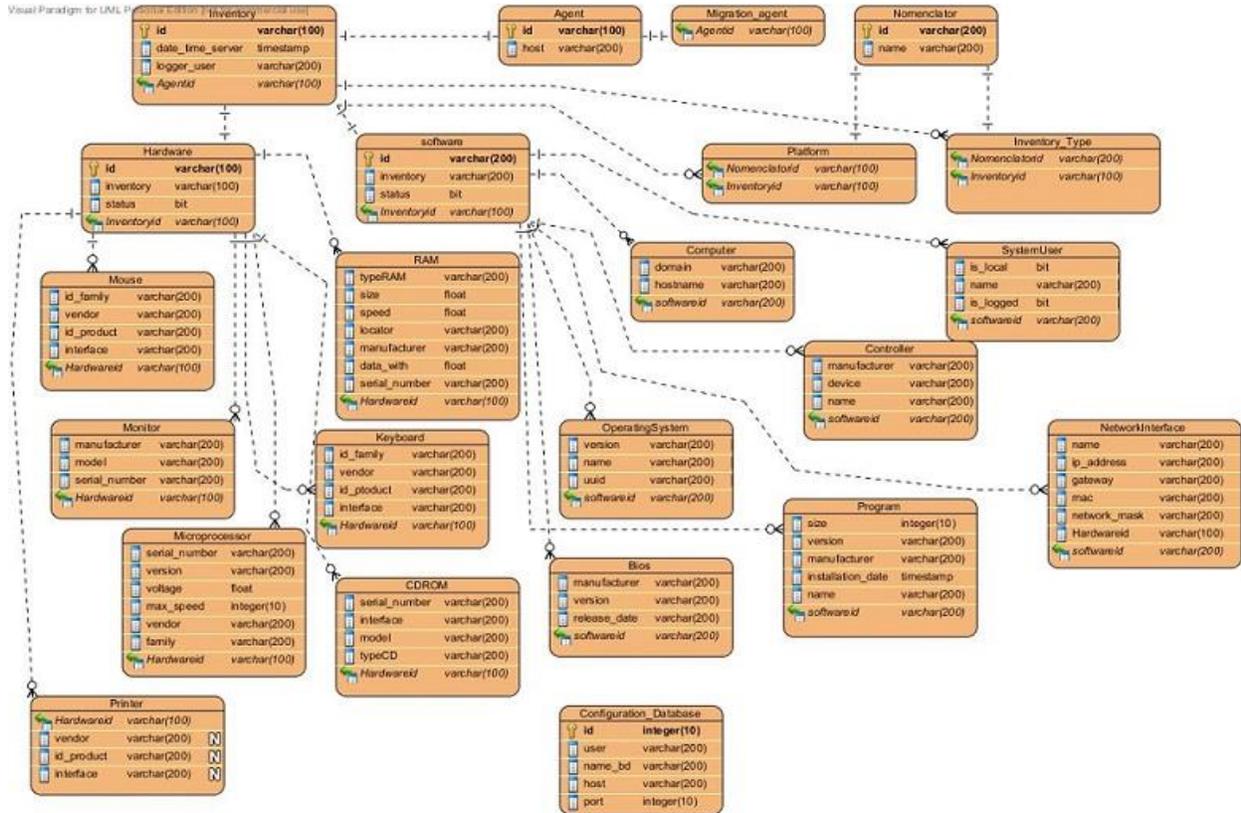


Figura 3: Diagrama Entidad Relación GRHS



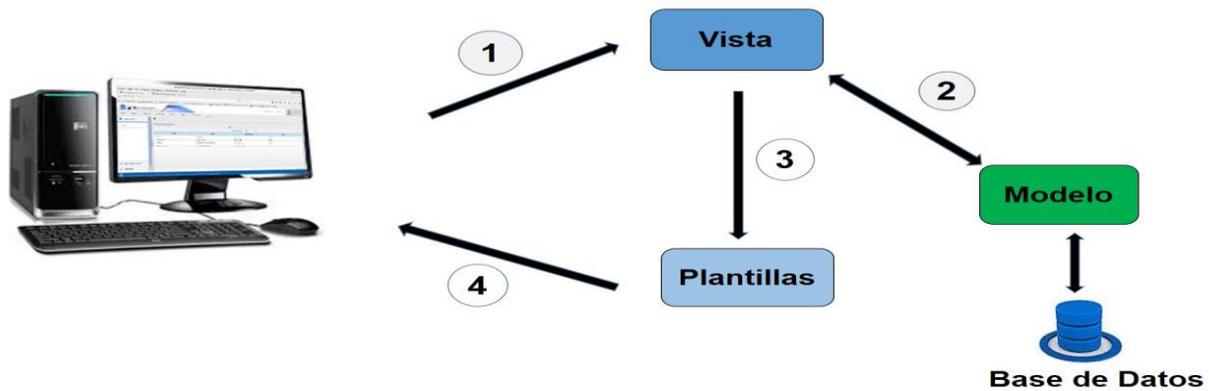
2.7.3 Patrón de arquitectura

Un patrón de arquitectura especifica un conjunto predefinido de subsistemas con sus responsabilidades y una serie de recomendaciones para organizar los distintos componentes. (38). El patrón arquitectónico que se utiliza es MTV el cual es la arquitectura usada en aplicaciones web, que hacen uso del framework Django, ya que fue adaptado en específico para este.

2.7.3.1 Arquitectura Modelo-Plantilla-Vista

El objetivo de utilizar patrones como el MTV es principalmente hacer más fluida la comunicación entre desarrolladores, ya que permite organizar el trabajo de los mismos, favorece a crear independencia en el funcionamiento, permitiendo realizar cambios en una parte en particular sin afectar el resto de la aplicación. Además facilita el mantenimiento en caso de errores y ofrece maneras más simples de demostrar el correcto desempeño de la aplicación. Separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. (39)

Figura 4: Arquitectura Modelo-Plantilla-Vista



1-El navegador manda una solicitud.

2-La vista interactúa con el modelo para obtener los datos.

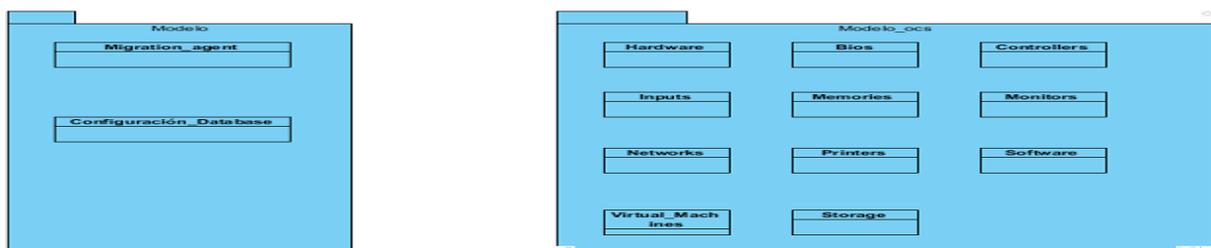
3-La vista envía la información a la plantilla.

4-La plantilla renderiza la respuesta a la solicitud del navegador.

Para representar como se evidencian estas capas en este trabajo de diploma se utilizan símbolos de los diagramas de clases.

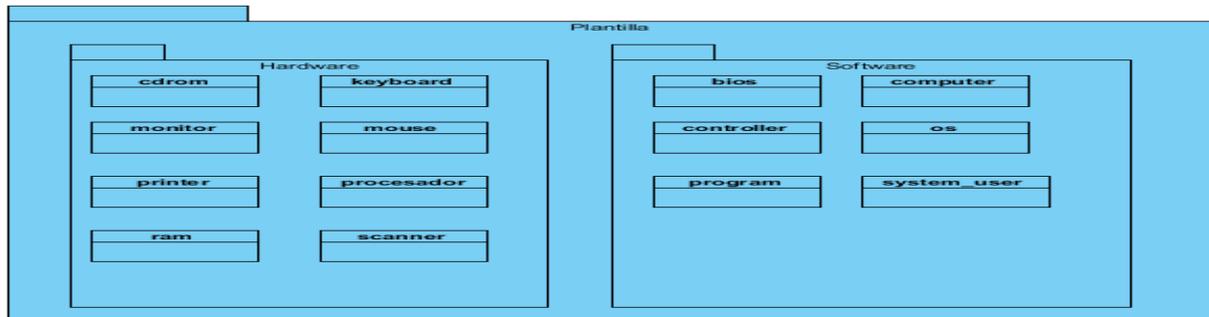
La Modelo: Se refiere a la capa de acceso de datos. Esta capa contiene todo lo respectivo a los datos, cómo acceder a ellos, cómo validarlos, qué comportamiento tienen y las relaciones entre ellos. En este caso este sistema utiliza dos sub-capas para representar el modelo, una llamada models que contiene la clase del formulario por donde se entran los datos para la conexión al SGBD MySQL y la otra llamada models_ocs que contiene los modelos correspondientes a la información que se migra situada en el SGBD MySQL. Ambas sub-capas se presentan a continuación.

Figura 5: Capa Modelo



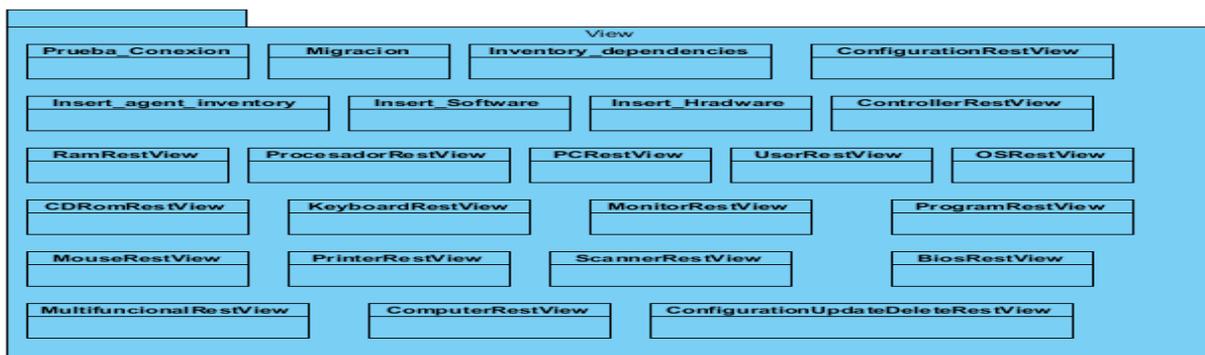
La Plantilla: Se refiere a la capa de presentación. Esta capa contiene las decisiones relacionadas con la presentación de cómo debe mostrarse la información en una página Web.

Figura 6: Capa Plantilla



La Vista: Se refiere a la capa de la lógica de negocio. Esta capa contiene el acceso al modelo y delega en la(s) plantilla(s) apropiada(s) esta información.

Figura 7: Capa Vista



2.7.4 Patrón de diseño

Los patrones de diseño son el esqueleto de las soluciones a problemas comunes en el desarrollo de software. (40)

2.7.4.1 Patrones GRASP

GRASP acrónimo que significa General Responsibility Assignment Software Patterns (Patrones de Software para la Asignación General de Responsabilidad). Las patrones GRASP constituyen la base del cómo se diseñará el sistema. “Describen los principios fundamentales de diseño de objetos y la asignación de responsabilidades, expresados como patrones”. (41). De los diferentes

Capítulo 2: Características del Sistema, Exploración, Planificación y Diseño

patrones que ofrece GRASP se ha tenido en cuenta para la modelación de la herramienta de migración de datos los siguientes:

Patrón Experto: El patrón garantiza que la responsabilidad de la creación de un objeto o la implementación de un método, recaiga sobre la clase que conoce toda la información necesaria para crearlo lo que contribuye a un adecuado encapsulamiento, favoreciendo la robustez y fácil mantenimiento del sistema. (41). En este caso la clase `Configuration_Migration` es la encargada de cumplir esta responsabilidad. Esta clase contiene la información necesaria para que se realice el proceso de migración.

Patrón Creador: Se asigna la responsabilidad a una clase de crear cuando contiene, agrega, compone, almacena o usa otra clase, lo que brinda una alta posibilidad de reutilizar la clase creadora. (41). En este caso las clases `Migration`, `Insert_Software`, `Insert_Hardware`, `Inventory_Dependencies` crean instancias de otras clases para realizar el proceso de migración.

Patrón Bajo Acoplamiento: Asigna una responsabilidad para mantener bajo acoplamiento. (41) Este da la medida de la poca dependencia que tiene una clase de otras clases. Las clases `Utils`, `Clean_Hardware` y `Clean_Software` no afectan los cambios en otras clases y se reutilizan en varias funcionalidades para transformar los datos que el gestor de destino de la migración PostgreSQL no asimile.

Patrón Alta Cohesión: Asigna responsabilidades de manera que una clase no tenga muchas funcionalidades no relacionadas o no realice un trabajo excesivo. Este patrón incrementa la claridad y facilita la comprensión del diseño. (41). Las clases `RamRestView`, `ProcesadorRestView`, `PCRestView`, `CDRomRestView`, `KeyboardRestView`, `ControllerRestView`, `OSRestView` son las clases donde se evidencia este patrón.

2.8 Conclusiones del capítulo

Durante el desarrollo de este capítulo se realiza la descripción de la solución propuesta, actividad que engloba un mapeo de bases de datos como respuesta a una detallada descripción del negocio. Además en este capítulo se identifican 6 HU las cuales satisfacen a los fundamentales procesos del negocio y se determinan las características no funcionales, vitales para el uso de la aplicación. También se elabora un plan de iteraciones, un plan de entrega, siguiendo las indicaciones de la fase de Planificación. Por último se evidencia el comportamiento de las clases en el módulo mediante el artefacto Tarjetas CRC, y se describen los patrones de diseño y de arquitectura que se utilizan.

CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBAS

3.1 Introducción

Para XP realizar un exitoso proceso de desarrollo orienta la ejecución de distintas fases las cuales en su realización generan los elementos o artefactos requeridos para la satisfacción del cliente en cuanto la calidad del software. En el presente capítulo se llevan a cabo las fases de implementación y pruebas. Se exponen las tareas de investigación correspondientes a determinada HU. Al mismo tiempo se exhibe un diagrama de despliegue capaz de modelar la arquitectura en tiempo de ejecución del módulo. De la misma forma se realizan pruebas en el módulo que se desarrolla con la intención de detectar errores, tanto por parte del cliente como del programador, para asegurar la calidad del producto.

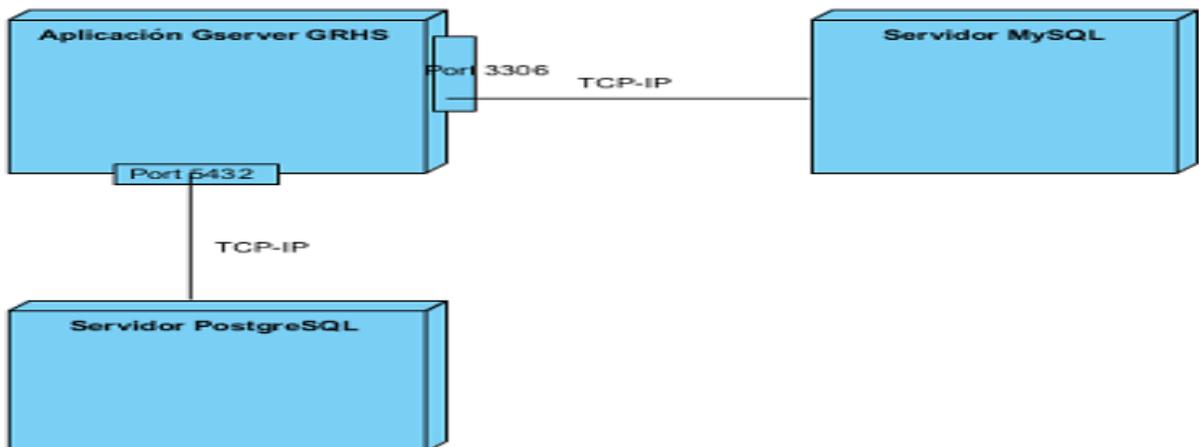
3.2 Implementación

La fase de Implementación de XP se caracteriza por la disponibilidad del cliente en contribuir a la calidad de lo que se implementa. Además se guía por los estándares de programación de acuerdo al lenguaje en uso. También determina que la implementación se debe realizar en parejas y el código es impersonal, es decir que ambos desarrolladores contribuyen a él de manera conjunta.

3.2.1 Diagrama de despliegue

El diagrama de despliegue modela la arquitectura en tiempo de ejecución de un sistema. Además es un UML que se utiliza para modelar el software y hardware utilizado en las implementaciones de sistemas y sus componentes. Por lo general los diagramas de despliegue se representan con nodos y componentes. (42)

Figura 8: Diagrama de despliegue



Aplicación Gserver: Módulo de GRHS que procesa la información recibida por los clientes donde se realiza el inventario de hardware y software.

Servidor MySQL: Se refiere al servidor MySQL en el cual son almacenados todos los datos recopilados por los inventarios de OCS Inventory NG.

Servidor PostgreSQL: Se refiere al servidor PostgreSQL en el cual son almacenados todos los datos recopilados por los inventarios de GRHS, y donde se va a insertar la información.

TCP-IP: Es la base del Internet que sirve para enlazar computadoras. El protocolo TCP/IP es utilizado para establecer la conexión entre el servidor de la aplicación y los servidores de bases de datos.

Port: Es el puerto por donde se conecta la aplicación Gserver a los servidores de base de datos donde se encuentra la información.

3.2.2 Estándar de nomenclatura y codificación en Python

Las convenciones o estándares de codificación son paradigmas de la programación que no están dirigidos a la lógica del sistema que se desarrolle, sino más bien a la estructura y apariencia física facilitando de esta manera la lectura, comprensión y mantenimiento del código. La metodología XP, la cual guía la realización de este trabajo, dicta al código como eje central para la comunicación de los programadores. Esto permite que durante la creación del producto se modifique cualquier parte del código existiendo previo acuerdo del dúo de programadores. Para cumplimentar esta práctica es primordial el uso de determinados estándares de codificación, los que le confieren al sistema una arquitectura y un estilo consistente. Los estándares resultan fácil de entender y mantener, y se basan en la legibilidad del código.

En Python existen los llamados PEP's⁶, entre los que se encuentra el PEP 8 el cual está dedicada a la recopilación de los estándares seguidos por los desarrolladores de Python a la hora de escribir código Python para la librería estándar. (43)

3.2.2.1 Comentarios

Resulta útil para los programadores situar comentarios en el código con el objetivo de esclarecer su función, facilitan el entendimiento del código y disminuyen la posibilidad de interpretarlo

⁶ Propuesta de mejoras en Python

erróneamente. Los comentarios deberán situarse de forma evidente, además deben ser claros y precisos, generalmente se usa un identificador. Es una necesidad primordial modificar los comentarios cada vez que se actualice el código. (43). En el trabajo se formulan comentarios sobre la línea de sentencia que se explica, luego de al menos dos espacios en blanco y un identificador. Además se utilizan comentarios en línea, que no es más que un comentario en la misma línea de una sentencia y con una separación en 2 espacios de esta, que comienza con un símbolo #.

3.2.2.2 Declaración de variables

En la declaración de variables se tiene en cuenta el estilo y las convenciones de nombrado de los lenguajes Python. Las variables son declaradas en una línea cada una de ellas. Empiezan con letra minúscula y se usa letra mayúscula para distinguir la letra inicial de palabras abreviadas y para relacionar términos se usa el guión bajo. Generalmente se conforman nombres sugerentes relacionados con el entorno del trabajo. (43)

3.2.2.3 Identificadores

Los identificadores sirven para nombrar variables, funciones y módulos. Pueden estar conformados por cualquiera de las 26 letras en minúsculas (a,.....,z) y los 10 dígitos (0 al 9). Se prescinde del uso de caracteres como ñ, ü, /, \$, &, etc debido a que no pueden ser leídos y entendidos correctamente en todos los entornos. (43)

3.2.2.4 Sentencias

Sentencias Simples

Cada línea está compuesta como máximo por una sentencia. Esclarecer que una sentencia puede ser la asignación de una función o de un objeto como literal, la declaración de un método o variable y el retorno de un método. (43)

Sentencias Compuestas

Se nombran sentencias compuestas a aquellas que contienen una lista de sentencias. (43). Estas están contenidas por estructuras de control como los If o For, los cuales se usan para acceder a toda la información conveniente para el proceso de migración.

3.2.2.5 Importaciones

Se realizan importaciones para recibir el auxilio de las funciones que brindan determinada clase, plugins o librería. Las importaciones deben estar en líneas separadas y se colocan al comienzo del archivo luego de cualquier comentario o documentación del módulo y antes de globales y constantes. (43)

3.2.3 Convenciones de codificación para JavaScript

Las convenciones de codificación del lenguaje de programación Java exponen reglas para el correcto uso del lenguaje de programación JavaScript. (44)

3.2.3.1 Nombres de fichero

Los ficheros se nombran siempre en minúsculas para evitar errores con aquellas plataformas case-sensitive⁷. Los ficheros tienen que concluir con la extensión .js, y no deben incluir signos de puntuación excepto – (guión medio) o _ (guión bajo), siendo preferible este último a la hora de separar palabras. (44)

3.2.3.2 Comentarios

Los comentarios deben escribirse de manera correcta y ser lo más esclarecedores posibles para que sean entendidos por las personas involucradas en el proceso de desarrollo. (44)

3.2.3.3 Declaración de variables

Las variables tienen que declararse utilizando la forma camelCase⁸. No deben ir precedidas de un guion bajo para indicar que se tratan de variables privadas. Cuando se declaren más de una variable de forma consecutiva, solo se utiliza una única sentencia var. Las variables que vayan a contener una instancia de JQuery, se identifican anteponiéndoles un signo de dólar (\$). (44)

3.2.3.4 Objeto global

El objeto global es aquel que envuelve a toda la aplicación y que evita la contaminación del entorno global. Para denotar su jerarquía, se declara en mayúsculas precedido del signo de dólar (\$). (44)

⁷ Expresión usada en informática que se aplica a los textos en los que tiene alguna relevancia escribir un carácter en mayúsculas o minúsculas.

⁸ Estilo de escritura que se aplica a frases o palabras compuestas.

3.2.3.5 Constructores

Los constructores (también llamados clases) tienen que nombrarse siguiendo la forma StudyCaps⁹ para diferenciarlos a simple vista del resto de estructuras. (44)

3.2.3.6 Métodos

Los métodos de los objetos tienen que ser nombrados utilizando camelCase, al igual que las variables y propiedades. (44)

3.2.4 Tareas de Ingeniería

Las Tareas de Ingeniería (TI) o de programación están basadas en la especificación por parte del programador de las HU en un lenguaje más técnico, para que este último entienda mejor la labor que tiene que hacer la funcionalidad, y pueda realizar una estimación más certera del esfuerzo en desarrollo que exige dicha implementación. Una HU puede generar varias TI. Se muestran seguidamente algunas de las TI que surgen para este trabajo y donde el desarrollo de cada una responde al funcionamiento de la HU pertinente. Las tareas de ingeniería son representadas por tablas divididas por las siguientes secciones:

- **Número Tarea:** Los números deben ser consecutivos.
- **Número Historia de Usuario:** Número de HU a la que da respuesta esta TI.
- **Nombre Tarea:** Nombre de la tarea.
- **Tipo de Tarea:** Las tareas pueden ser de: Desarrollo, Corrección y Mejora.
- **Puntos Estimados:** Duración de la realización de la TI de una manera más certera, el tiempo estimado es en semanas.
- **Programador responsable:** Nombre del programador.
- **Descripción:** Breve descripción de la tarea.

⁹ Estilo de escritura que se aplica a frases o palabras compuestas.

Tabla 9: TI # 1 Migrar información obtenida de la base de datos de OCS Inventory NG

Tareas de Ingeniería	
Número Tarea: 1	Número Historia de Usuario: 1
Nombre Tarea: Migrar información obtenida de la base de datos de OCS Inventory NG	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1
Programador responsable: Ronal Ledesma López y Bladimir Almeida Rodríguez	
Descripción: El sistema emplea la clase Migracion la cual guía el proceso de migración, mediante el apoyo en las clases que realizan la extracción, transformación e inserción en la base de datos de GRHS. El procedimiento para realizar esta operación se basa en la extracción de todos los host existente en la BD de OCS Inventory NG, luego se insertan en la BD de GRHS, y a partir de ahí son migrados la información correspondiente a software y hardware referente a cada host.	

Tabla 10: TI # 5 Adicionar Conexión a MySQL

Tareas de Ingeniería	
Número Tarea: 5	Número Historia de Usuario: 2
Nombre Tarea: Adicionar conexión a MySQL	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1
Programador responsable: Ronal Ledesma López	
Descripción: El usuario selecciona la funcionalidad (Adicionar), luego se muestra una formulario con los datos requeridos para guardar una posible conexión. Estos datos son enviados a través del método DATA.	

El resto de las TI se encuentra en el **Anexo IV** del proyecto.

3.3 Pruebas

Las pruebas representan un interesante reto para los ingenieros de software, quienes por su naturaleza son personas constructivas. Las pruebas requieren que el desarrollador descarte nociones preconcebidas de lo que es correcto en el software y entonces diseñe difíciles casos de prueba para romperlo. (45)

Una estrategia de prueba del software proporciona una descripción o planificación bien detallada de los pasos a llevar a cabo como parte de la prueba que da como resultado una correcta construcción del software. Esta debe incluir pruebas de bajo nivel que verifiquen que todos los pequeños segmentos de código fuente se han implementado correctamente, así como pruebas de alto nivel que validen las principales funciones del sistema frente a los requerimientos del cliente. La prueba es aplicada para diferentes tipos de objetivos, en diferentes escenarios o niveles de trabajo. En este caso se aplican los siguientes niveles de prueba:

- Prueba de unidad.
- Prueba de integración.
- Prueba de aceptación.

3.3.1 Métodos de prueba

Son dos fundamentales: el método de la caja blanca y de la caja negra.

La prueba de la **caja blanca** del software se comprueba los caminos lógicos del software proponiendo casos de prueba que se ejerciten conjuntos específicos de condiciones y/o bucles. Se puede examinar el estado del programa en varios puntos para determinar si el estado real coinciden con el esperado o mencionado. (46)

La prueba de **caja negra** se refiere a las pruebas que se llevan a cabo sobre la interfaz del software. O sea, los casos de prueba pretenden demostrar que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se produce un resultado correcto, así como que la integridad de la información externa se mantiene. (46)

3.3.1.1 Técnicas de prueba de caja blanca

La prueba del camino básico: Esta prueba permite al diseñador de casos de prueba obtener una medida de la complejidad lógica de un diseño procedimental y usar esa medida como guía para la definición de un conjunto básico de caminos de ejecución.

La prueba de condición: Es un método de diseño de casos de prueba que ejercita las condiciones lógicas contenidas en el módulo de un programa.

La prueba de flujo de datos: Se selecciona caminos de prueba de un programa de acuerdo con la ubicación de las definiciones y los usos de las variables del programa.

La prueba de bucles: Es una técnica de prueba de caja blanca que se centra exclusivamente en la validez de las construcciones de bucles. (46)

3.3.1.2 Automatización de las técnicas de prueba de caja blanca

El proceso de prueba es necesario realizarlo en todo el ciclo de vida del desarrollo del software. Esperar a realizar el desarrollo del software para luego efectuar la prueba se constituye en una práctica de alto riesgo. Esto se debe a que las fallas encontradas al desarrollar el software pueden ser muchas y difíciles de remover, ya que se está probando todo el sistema y es difícil saber dónde se encuentra el defecto. Por esto se realiza la prueba a todo nivel, **unitaria**, de **integración** y de **sistema**. (46)

3.3.2 Pruebas unitaria

La prueba de unidad se concentra en el esfuerzo de verificación de la unidad más pequeña de diseño del software: el componente o módulo del software. (45). Estas pruebas se concentran en la lógica del procesamiento interno y en las estructuras de datos dentro de los límites de un componente y prueban todos los caminos de control para descubrir errores dentro de esos límites. Son desarrolladas a nivel de unidad y fijan su atención en la validación de funciones y métodos que responden a requisitos del sistema.

Mediante ellas se verifica el correcto funcionamiento de las unidades lógicas en las que se divide el software, sin tener en cuenta las interrelaciones con otras unidades. Para la ejecución de las pruebas unitarias se utiliza la herramienta unittest, también llamada PYunit, contenida en el lenguaje de programación Python. Esta forma parte de un conjunto de herramientas llamadas xUnit, que tuvo como creador a Kent Beck. (47)

A continuación se muestran los resultados obtenidos durante la aplicación de dichas pruebas al módulo que se desarrolla. Se realizan 4 iteraciones para 17 fragmentos de código, sobre los que se detectan 6 errores que se resuelven de manera satisfactoria. La pruebas que se realizan pueden consultarse en el archivo tests.py del módulo migración.

3.3.2.1 Iteración I

En esta iteración se seleccionan 5 fragmentos de código en los cuales se detectan 3 errores de lógica algorítmica, que se resuelven de manera satisfactoria.

Figura 9: Resultados de las pruebas unitarias. I Iteración

```
-----  
Ran 5 tests in 0.038s  
  
OK  
Destroying test database for alias 'default'...  
Destroying test database for alias 'ocs'...
```

3.3.2.2 Iteración II

En esta iteración se seleccionan 4 fragmentos de código en los cuales se detectan 2 errores de lógica algorítmica, que se resuelven de manera satisfactoria.

Figura 10: Resultados de las pruebas unitarias. II Iteración

```
-----  
Ran 4 tests in 0.010s  
  
OK  
Destroying test database for alias 'default'...  
Destroying test database for alias 'ocs'...
```

3.3.2.3 Iteración III

En esta iteración se seleccionan 7 fragmentos de código en el cuál se detectan 3 errores. Los errores devuelven que los datos estaban mal serializados por lo que se soluciona serializándolos de la forma correcta.

Figura 11: Resultados de las pruebas unitarias. III Iteración

```
-----  
Ran 7 tests in 0.060s  
  
OK  
Destroying test database for alias 'default'...  
Destroying test database for alias 'ocs'...
```

3.3.2.4 Iteración IV

En esta iteración se selecciona 1 fragmento de código el cuál devuelve que está mal serializado el valor. Lo que se soluciona serializándolo correctamente.

Figura 12: Resultados de pruebas unitarias. IV Iteración

```
-----  
Ran 1 test in 0.001s  
  
OK  
Destroying test database for alias 'default'...  
Destroying test database for alias 'ocs'...
```

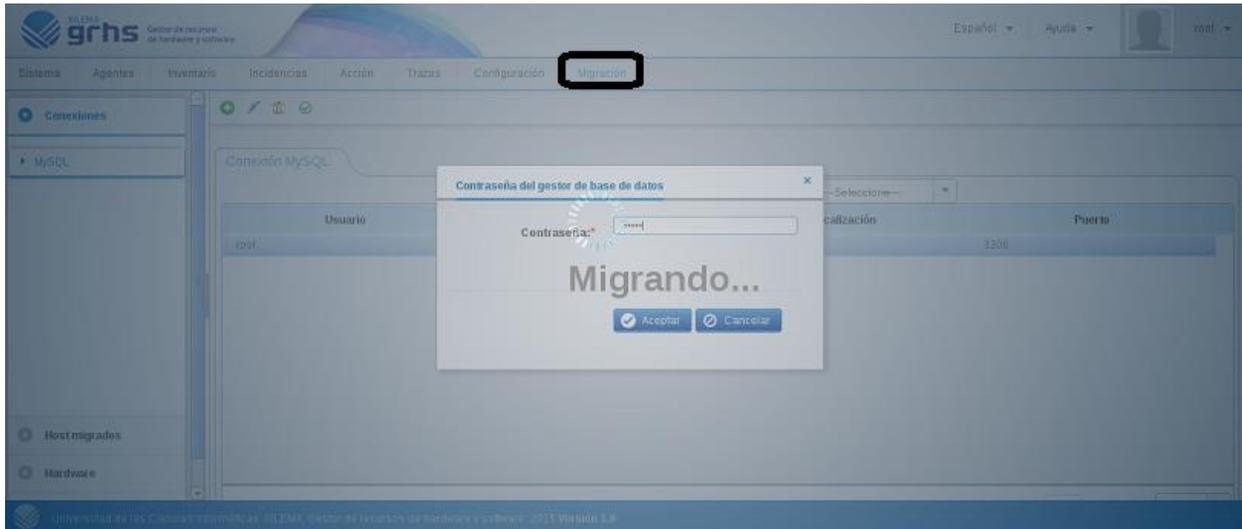
3.3.3 Pruebas de integración

Las pruebas de integración orientadas a objetos se enfocan a la interacción entre unidades, suponiendo que cada una fue probada a nivel de unidad. A este nivel se mezclan aspectos estructurales que relacionan las maneras de interactuar de las unidades y también los aspectos típicamente funcionales. (48) Las pruebas de integración se llevan a cabo durante la construcción del sistema, involucran a un número creciente de módulos y terminan probando el sistema como conjunto. Estas pruebas se pueden plantear desde un punto de vista estructural o funcional. Las pruebas estructurales de integración son similares a las pruebas de caja blanca; pero trabajan a un nivel conceptual superior. En lugar de referirse a sentencias del lenguaje, se refiere a llamadas entre módulos. Se trata de identificar todos los posibles esquemas de llamadas y ejercitarlos para lograr una buena cobertura de segmentos o de ramas. Las pruebas funcionales de integración son similares a las pruebas de caja negra. En este espacio se pretende encontrar los fallos en la respuesta de un módulo cuando su operación depende de los servicios prestados por otro(s) módulo(s). (49)

En esta investigación se aplican las pruebas de integración desde una perspectiva funcional. La inclusión del módulo Migración al sistema GRHS ha sido satisfactoria, cumple con la especificación de requisitos del usuario.

Desde el punto de vista estructural la prueba de integración responden al método de prueba de caja blanca. En este caso la prueba de integración que se realiza devuelve una no conformidad. Esta consiste en la utilización y modificación de una librería perteneciente al sistema donde se va a integrar el módulo, lo cual no favorece a la utilización de este en cualquier nueva versión del gserver. Esta no conformidad se soluciona satisfactoriamente creando un archivo perteneciente a este módulo Migración donde se personalizan los formularios y botones requeridos para dicho módulo.

Figura 13: Muestra de la prueba de integración



3.3.4 Pruebas de aceptación

Las pruebas de aceptación pertenecen al método de caja negra y se realizan para evaluar el grado de calidad del software de acuerdo a todos los aspectos relevantes que intervienen en el sistema, y de esa manera justificar el uso del producto.

Al llevar cabo las pruebas de aceptación para el módulo de migración desarrollado, se obtuvieron 13 no conformidades que engloban los tipos de diseño de interfaz y la internacionalización del módulo, las cuáles fueron resueltas en su totalidad. La siguiente gráfica funge como evidencia de esto.

Figura 14: Gráfico que recoge el comportamiento de las no conformidades planteadas por el usuario en las pruebas de aceptación

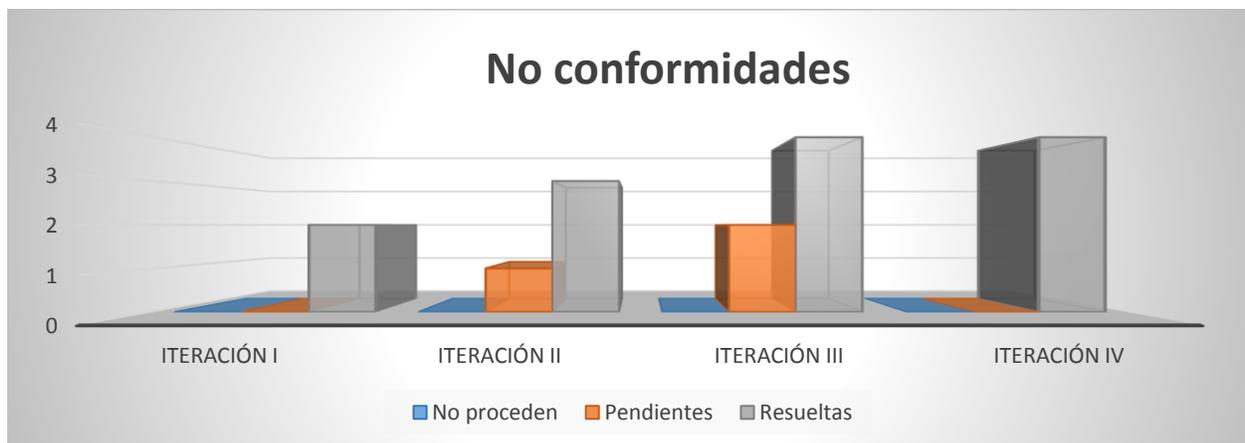


Tabla 11: Prueba de aceptación # 1 HU Gestionar conexiones

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observación
El usuario accede al módulo (Migración), y selecciona la opción (Conexiones), luego escoge el botón (Adicionar) e introduce correctamente los campos requeridos.		El sistema muestra un mensaje “La operación se realizó con éxito”.	Satisfactorio	Adicionar una conexión no significa que se establece una conexión, sino que se guardan los datos para posteriormente establecer la conexión
	El usuario accede al módulo (Migración), y selecciona la opción (Conexiones), luego escoge el botón (Adicionar) y no introduce todos los campos.	El sistema muestra un error “Por favor inserte un valor(es) para este campo(s)”	No se adicionan los datos de la conexión	
	El usuario accede al módulo (Migración), y	El sistema muestra un error “Introduzca una	No se adicionan los datos de la conexión.	

Capítulo 3: Implementación y Pruebas

	selecciona la opción (Conexiones), luego escoge el botón (Adicionar) e introduce datos en formato desconocido para el campo (Localización).	dirección IPv4 o IPv6 válida”.		
	El usuario accede al módulo (Migración) y selecciona la opción (Conexiones), pero el sistema no tiene conexión con el servidor.	El sistema no responde.	No se muestra la información de las conexiones	

El resto de las pruebas de aceptación realizadas se pueden consultar en el **Anexo V**.

3.5 Conclusiones del Capítulo

En el desarrollo de este capítulo se plantea un diagrama de despliegue para modelar la arquitectura del sistema en tiempo de ejecución. Se realiza cada una de las tareas de ingeniería antes mencionada, para dar respuesta a las historias de usuario y concretar el sistema propuesto. Se aplican los niveles de pruebas, unitarias, integración y de aceptación para verificar la funcionalidad del módulo.

Conclusiones generales

El trabajo describe en gran nivel de detalles la herramienta realizada para la migración de datos desde el sistema OCS Inventory NG 2.0.5 a GRHS1.0. Una vez finalizada la investigación se concluye que:

- El estudio de los sistemas para la migración de datos permite definir la factibilidad de la propuesta. Los sistemas existentes no cubren los requisitos que impone el negocio de GRHS para el proceso de migración entre las BD de OCS Inventory NG y este.
- La selección de las herramientas utilizadas en el desarrollo del módulo está orientada a las necesidades y restricciones del cliente.
- El módulo desarrollado puede ser aprovechado en toda aplicación que soporte el framework Xilema-Base-Web. Permite que se migre toda la información manejada de forma similar en las herramientas GRHS 1.0 y OCS Inventory NG 2.0.5.
- Se obedecen las exigencias de XP acerca de las fases y prácticas a efectuar para lograr un producto de calidad, obteniendo esta condición mediante los artefactos propuestos por la metodología.

A partir del análisis anterior se expone que el presente trabajo de diploma alcanza los objetivos propuestos.

Recomendaciones

Luego de haber analizado los resultados del presente trabajo de diploma se recomienda:

- Estudiar la posibilidad de agregar a este módulo nuevas funcionalidades capaces de que el sistema pueda migrar información para una nueva versión de OCS Inventory NG en caso de que cambie la estructura de la base de datos de esta con respecto a la versión 2.0.5.
- Incorporar a este módulo la funcionalidad que se encargue de exportar la información generada luego de la migración a formato Excel con el fin de que se recoja en este tipo de documentos la información migrada, para satisfacer las exigencias de determinado cliente.

Referencias

1. Jose Antonio Hernandez, Yoanni Ordoñez, Ernesto Aviles Vazquez. *GESTIÓN DE INCIDENCIAS EN INVENTARIOS DE RED*. Universidad de las Ciencias Informáticas. La Habana : s.n., 2013.
2. OCS Inventory NG. *OCS Inventory NG*. [En línea] OCS Inventory NG Team, 2001-2014. <http://www.ocsinventory-ng.org/en/about/>.
3. Xilema GRHS. *Xilema GRHS*. [En línea] Universidad de la Ciencias Informáticas, 2015. [Citado el: 5 de junio de 2015.] <http://10.128.50.236/grhs-doc/index.php/GRHS>.
4. Correa, Susana Laura Corona. *Factores críticos de éxitos en el proceso de migración de base de datos*. Dirección General de Servicios de Cómputo Académico, Universidad Autónoma de México. México : s.n.
5. KPMG. *KPMG*. [En línea] 2015. [Citado el: 8 de diciembre de 2014.] <http://www.kpmg.com/CO/es/IssuesAndInsights/ArticlesPublications/Documents/Portafolio%20de%20Productos%20de%20KPMG%20-%20ENTERPRISE%20INFORMATIO%20MANAGEMENT.pdf>.
6. Slideshare. *Slideshare*. [En línea] 2015. [Citado el: 14 de diciembre de 2014.] <http://es.slideshare.net/sadow/qu-es-un-dato-8210460>.
7. C.J.Date. *Introducción a los sistemas de bases de datos*. México : Pearson Educación, 2001.
8. Rodríguez, Juan Ramón López. Laboratorio de Bases de Datos. *Laboratorio de Bases de Datos*. [En línea] 2010. [Citado el: 14 de diciembre de 2014.] <http://docencia.lbd.udc.es/bdd/teoria/tema2/2.3.1.-ElModeloRelacional.pdf>.
9. Ing. Karel Rodríguez Carmenates, Ing. Félix González Martínez. Fordes. *Fordes*. [En línea] noviembre de 2014. [Citado el: 21 de noviembre de 2014.] <http://semanatecnologica.fordes.co.cu/?q=node/1552>.
10. DBConvert. *DBConvert*. [En línea] 2001-2014. [Citado el: 14 de noviembre de 2014.] <http://dbconvert.com/convert-mysql-to-postgresql-pro.php>.
11. PostgreSQL. *PostgreSQL*. [En línea] 1996-2015. [Citado el: 26 de noviembre de 2014.] <http://archives.postgresql.org/pgsql-es-ayuda/2009-01/msg00375.php>.

12. Soft32Download. *Soft32Download*. [En línea] 2006-2015. [Citado el: 20 de noviembre de 2014.] [http://my2pg.script.soft32download.com/..](http://my2pg.script.soft32download.com/)
13. Intelligent Converters. *Intelligent Converters*. [En línea] 2001-2015. [Citado el: 21 de noviembre de 2014.] <http://www.convert-in.com/sql2pgs.htm>.
14. Navicat. *Navicat*. [En línea] 1999-2015. [Citado el: 25 de noviembre de 2014.] <http://www.navicat.com/es/whatisnavicat>.
15. Ing. Karel Rodríguez Carmenates, Ing. Félix González Martínez. Fordes. *Fordes*. [En línea] noviembre de 2014. [Citado el: 30 de noviembre de 2014.] <http://semanatecnologica.fordes.co.cu/?q=node/1552>.
16. Pressman, Roger. *Ingeniería de Software: Un enfoque práctico, 7ma edición*. Connecticut : McGraw Hill.
17. Patricio Letelier, María del Carmen Panadés. *Metodologías Ágiles para el desarrollo de Software: Extreme Programming (XP)*. Departamento de Sistemas Informáticos y Computación, Universidad Politécnica de Valencia. Valencia : s.n.
18. Duque, Raul González. *Python para todos*. España : Creative Commons Reconocimiento.
19. Pérez, Javier Eguíluz. Libros Web CSS. *Libros Web CSS*. [En línea] 2005-2014. [Citado el: 20 de noviembre de 2014.] http://www.librosweb.es/css/pdf/introduccion_css.pdf.
20. Pilgrim, Mark. *HTML5 Up and Running*. s.l. : O'Reilly Media, 2010.
21. Pérez, Javier Eguíluz. Libros Web Introducción a JavaScript. *Libros Web Introducción a JavaScript*. [En línea] 2005-2014. [Citado el: 22 de noviembre de 2014.] <http://librosweb.es/libro/javascript/>.
22. PostgreSQL Documentation. *PostgreSQL Documentation*. [En línea] universidad de las Ciencias Informáticas, 2006-2015. [Citado el: 14 de diciembre de 2014.] <http://docs.prod.uci.cu/online/PostgreSQL.docset/Contents/Resources/Documents/preface.html>.
23. Gilfillan, Ian. *Labiblia de MySQL*. s.l. : Anaya Multimedia, 2003.

24. pgAdmin PostgreSQL Tools. *pgAdmin PostgreSQL Tools*. [En línea] 2002 - 2009. [Citado el: 10 de diciembre de 2014.] <http://www.pgadmin.org/>.
25. phpMyAdmin. *phpMyAdmin*. [En línea] 2012-2014. <http://docs.phpmyadmin.net/en/latest/>.
26. Visual Paradigm. *Visual Paradigm*. [En línea] 16 de agosto de 2010. [Citado el: 2014 de 12 de 6.] <http://www.visual-paradigm.com/support/vpum/releasenotes/800.jsp>.
27. Systems, Popkin Software and. TLDP/ES Lucas. *TLDP/ES Lucas*. [En línea] 2007. [Citado el: 15 de Enero de 2015.] <http://es.tldp.org/Tutoriales/doc-modelado-sistemas-UML/doc-modelado-sistemas-uml.pdf>.
28. Pérez, Javier Eguíluz. Libros Web. . *Libros Web*. [En línea] 2005-2014. [Citado el: 26 de noviembre de 2014.] <http://librosweb.es/libro/Ajax/>.
29. Ecma International. *Ecma International*. [En línea] Octubre de 2013. [Citado el: 26 de noviembre de 2014.] <http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf>.
30. Xilema GRHS . *Xilema GRHS* . [En línea] Centro de telemática, 2015. [Citado el: 5 de diciembre de 2014.] http://10.128.50.236/grhs-doc/index.php/Plugin_para_Xilema_Base_Web.
31. Moss, Adrian Holovaty y Jacob Kaplan. *Django Book*. s.l. : Jeremy Dunck, 2008.
32. backbonejs. *backbonejs*. [En línea] 2015. [Citado el: 2014 de diciembre de 2015.] <http://backbonejs.org>.
33. Desarrollo Web. *Desarrollo Web*. [En línea] 2014. [Citado el: 2014 de diciembre de 5.] <http://www.desarrolloweb.com/manuales/manual-jquery.html>.
34. Pincirolí, Lic. Fernando. SEDICI. *SEDICI*. [En línea] 2003-2015. [Citado el: 22 de enero de 2015.] <http://sedici.ulnp.edu.ar/handle/10915/23050>.
35. Panadés, Patricio Letelier y María del Carmén. *Metodologías Ágiles para el desarrollo de software*. Departamento de Sistemas Informáticos y Computación, Universidad Politécnica de Valencia. 2012.

36. Pressman, Roger. *Ingeniería del Software: Un enfoque práctico*. Connecticut : Mc Graw Hill, 2005.
37. Pinciroli, Lic. Fernando. SEDICI. *SEDICI*. [En línea] 2003-2015. [Citado el: 2 de febrero de 2015.] <http://sedici.unlp.edu.ar/handle/10915/23050>.
38. Departamento de Lenguajes y Sistemas Informáticos. *Departamento de Lenguajes y Sistemas Informáticos*. [En línea] 2015. [Citado el: 5 de febrero de 2015.] <http://www.lsi.us.es/docencia/get.php?id=1130>.
39. Moss, Adrian Holovaty y Jacob Kaplan. *Django Book*. [ed.] Jeremy Hunck. 2008.
40. Developer Network. *Developer Network*. [En línea] 2015. [Citado el: 5 de marzo de 2015.] <https://msdn.microsoft.com/es-es/library/bb972240.aspx>.
41. Astudillo, Marcello Visconti y Hernán. *Fundamentos de Ingeniería de Software*. Departamento de Informática, Universidad Técnica Federico Santa María.
42. Slideshare. *Slideshare*. [En línea] 2015. [Citado el: 1 de marzo de 2015.] http://es.slideshare.net/joshell/diagramas-uml-componentes-y-despliegue?next_slideshow=1.
43. Recursos de Python. *Recursos de Python*. [En línea] 2015. [Citado el: 5 de marzo de 2015.] <http://recursospython.com/pep8es.pdf>.
44. etnassoft. *etnassoft*. [En línea] 2011. [Citado el: 6 de marzo de 2015.] <http://www.etnassoft.com/2012/10/23/guia-de-estilo-nomenclatura-en-archivos-javascript/>.
45. Pressman, Roger. *Técnicas de Prueba de Software*. [aut. libro] Roger Presman. *Ingeniería del Software: Un enfoque práctico*. Connecticut : McGraw Hill.
46. Informáticas, Departamento de Ingeniería del Software Facultad 2 Universidad de las Ciencias. *Conferencia acerca de las pruebas de software*. 2014.
47. Duque, Raúl González. *Python para todos*. España : Creative Commons Reconocimiento.
48. Universidad Veracruzana. *Universidad Veracruzana*. [En línea] 2015. <http://www.uv.mx/personal/jfernandez/files/2010/07/Pruebas-de-Integracion.pdf>.

49. Martínez, Ing Eduardo Salazar. Portal de e-governo, inclusão digital e sociedade do conhecimento. *Portal de e-governo, inclusão digital e sociedade do conhecimento*. [En línea] 2015. [Citado el: 25 de mayo de 2015.] <http://www.egov.ufsc.br/portal/conteudo/propuesta-de-procedimiento-para-realizar-pruebas-de-caja-blanca-las-aplicaciones-que-se-des>.

Bibliografía

1. Jose Antonio Hernandez, Yoanni Ordoñez, Ernesto Aviles Vazquez. *GESTIÓN DE INCIDENCIAS EN INVENTARIOS DE RED*. Universidad de las Ciencias Informáticas. La Habana : s.n., 2013.
2. OCS Inventory NG. *OCS Inventory NG*. [En línea] OCS Inventory NG Team, 2001-2014. <http://www.ocsinventory-ng.org/en/about/>.
3. Xilema GRHS. *Xilema GRHS*. [En línea] Universidad de la Ciencias Informáticas, 2015. [Citado el: 5 de junio de 2015.] <http://10.128.50.236/grhs-doc/index.php/GRHS>.
4. Correa, Susana Laura Corona. *Factores críticos de éxitos en el proceso de migración de base de datos*. Dirección General de Servicios de Cómputo Académico, Universidad Autónoma de México. México : s.n.
5. KPMG. *KPMG*. [En línea] 2015. [Citado el: 8 de diciembre de 2014.] <http://www.kpmg.com/CO/es/IssuesAndInsights/ArticlesPublications/Documents/Portafolio%20de%20Productos%20de%20KPMG%20-%20ENTERPRISE%20INFORMATIO%20MANAGEMENT.pdf>.
6. Slideshare. *Slideshare*. [En línea] 2015. [Citado el: 14 de diciembre de 2014.] <http://es.slideshare.net/sadow/qu-es-un-dato-8210460>.
7. C.J.Date. *Introducción a los sistemas de bases de datos*. México : Pearson Educación, 2001.
8. Rodríguez, Juan Ramón López. Laboratorio de Bases de Datos. *Laboratorio de Bases de Datos*. [En línea] 2010. [Citado el: 14 de diciembre de 2014.] <http://docencia.lbd.udc.es/bdd/teoria/tema2/2.3.1.-ElModeloRelacional.pdf>.
9. Ing. Karel Rodríguez Carmenates, Ing. Félix González Martínez. Fordes. *Fordes*. [En línea] noviembre de 2014. [Citado el: 21 de noviembre de 2014.] <http://semanatecnologica.fordes.co.cu/?q=node/1552>.
10. DBConvert. *DBConvert*. [En línea] 2001-2014. [Citado el: 14 de noviembre de 2014.] <http://dbconvert.com/convert-mysql-to-postgresql-pro.php..>
11. PostgreSQL. *PostgreSQL*. [En línea] 1996-2015. [Citado el: 26 de noviembre de 2014.] <http://archives.postgresql.org/pgsql-es-ayuda/2009-01/msg00375.php..>

12. Soft32Download. *Soft32Download*. [En línea] 2006-2015. [Citado el: 20 de noviembre de 2014.] [http://my2pg.script.soft32download.com/..](http://my2pg.script.soft32download.com/)
13. Intelligent Converters. *Intelligent Converters*. [En línea] 2001-2015. [Citado el: 21 de noviembre de 2014.] <http://www.convert-in.com/sql2pgs.htm>.
14. Navicat. *Navicat*. [En línea] 1999-2015. [Citado el: 25 de noviembre de 2014.] <http://www.navicat.com/es/whatisnavicat>.
15. Pressman, Roger. *Ingeniería de Software: Un enfoque práctico, 7ma edición*. Connecticut : McGraw Hill.
16. Duque, Raul González. *Python para todos*. España : Creative Commons Reconocimiento.
17. Pérez, Javier Eguíluz. Libros Web CSS. *Libros Web CSS*. [En línea] 2005-2014. [Citado el: 20 de noviembre de 2014.] http://www.librosweb.es/css/pdf/introduccion_css.pdf.
18. Pilgrim, Mark. *HTML5 Up and Running*. s.l. : O'Reilly Media, 2010.
19. Pérez, Javier Eguíluz. Libros Web Introducción a JavaScript. *Libros Web Introducción a JavaScript*. [En línea] 2005-2014. [Citado el: 22 de noviembre de 2014.] <http://librosweb.es/libro/javascript/>.
20. PostgreSQL Documentation. *PostgreSQL Documentation*. [En línea] universidad de las Ciencias Informáticas, 2006-2015. [Citado el: 14 de diciembre de 2014.] <http://docs.prod.uci.cu/online/PostgreSQL.docset/Contents/Resources/Documents/preface.html>.
21. Gilfillan, Ian. *Labiblia de MySQL*. s.l. : Anaya Multimedia, 2003.
22. pgAdmin PostgreSQL Tools. *pgAdmin PostgreSQL Tools*. [En línea] 2002 - 2009. [Citado el: 10 de diciembre de 2014.] <http://www.pgadmin.org/>.
23. phpMyAdmin. *phpMyAdmin*. [En línea] 2012-2014. <http://docs.phpmyadmin.net/en/latest/>.
24. Visual Paradigm. *Visual Paradirgm*. [En línea] 16 de agosto de 2010. [Citado el: 2014 de 12 de 6.] <http://www.visual-paradigm.com/support/vpum/releasenotes/800.jsp>.

25. Systems, Popkin Software and. *TLDP/ES Lucas*. *TLDP/ES Lucas*. [En línea] 2007. [Citado el: 15 de Enero de 2015.] <http://es.tldp.org/Tutoriales/doc-modelado-sistemas-UML/doc-modelado-sistemas-uml.pdf>.
26. Pérez, Javier Eguíluz. *Libros Web*. . *Libros Web*. [En línea] 2005-2014. [Citado el: 26 de noviembre de 2014.] <http://librosweb.es/libro/Ajax/>.
27. Ecma International. *Ecma International*. [En línea] Octubre de 2013. [Citado el: 26 de noviembre de 2014.] <http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf>.
28. Xilema GRHS . *Xilema GRHS* . [En línea] Centro de telemática, 2015. [Citado el: 5 de diciembre de 2014.] http://10.128.50.236/grhs-doc/index.php/Plugin_para_Xilema_Base_Web.
29. backbonejs. *backbonejs*. [En línea] 2015. [Citado el: 2014 de diciembre de 2015.] <http://backbonejs.org>.
30. Desarrollo Web. *Desarrollo Web*. [En línea] 2014. [Citado el: 2014 de diciembre de 5.] <http://www.desarrolloweb.com/manuales/manual-jquery.html>.
31. Pincioli, Lic. Fernando. *SEDICI*. *SEDICI*. [En línea] 2003-2015. [Citado el: 22 de enero de 2015.] <http://sedici.ulnp.edu.ar/handle/10915/23050>.
32. Panadés, Patricio Letelier y María del Carmén. *Metodologías Ágiles para el desarrollo de software*. Departamento de Sistemas Informáticos y Computación, Universidad Politécnica de Valencia. 2012.
33. Pressman, Roger. *Ingeniería del Software: Un enfoque práctico*. Connecticut : Mc Graw Hill, 2005.
34. Departamento de Lenguajes y Sistemas Informáticos. *Departamento de Lenguajes y Sistemas Informáticos*. [En línea] 2015. [Citado el: 5 de febrero de 2015.] <http://www.lsi.us.es/docencia/get.php?id=1130>.
35. Moss, Adrian Holovaty y Jacob Kaplan. *Django Book*. [ed.] Jeremy Hunck. 2008.
36. Developer Network. *Developer Network*. [En línea] 2015. [Citado el: 5 de marzo de 2015.] <https://msdn.microsoft.com/es-es/library/bb972240.aspx>.

37. Astudillo, Marcello Visconti y Hernán. *Fundamentos de Ingeniería de Software*. Departamento de Informática, Universidad Técnica Federico Santa María.
38. Slideshare. *Slideshare*. [En línea] 2015. [Citado el: 1 de marzo de 2015.] http://es.slideshare.net/joshell/diagramas-uml-componentes-y-despliegue?next_slideshow=1.
39. Recursos de Python. *Recursos de Python*. [En línea] 2015. [Citado el: 5 de marzo de 2015.] <http://recursospython.com/pep8es.pdf>.
40. etnassoft. *etnassoft*. [En línea] 2011. [Citado el: 6 de marzo de 2015.] <http://www.etnassoft.com/2012/10/23/guia-de-estilo-nomenclatura-en-archivos-javascript/>.
41. Pressman, Roger. *Técnicas de Prueba de Software*. [aut. libro] Roger Presman. *Ingeniería del Software: Un enfoque práctico*. Connecticut : McGraw Hill.
42. Informáticas, Departamento de Ingeniería del Software Facultad 2 Universidad de las Ciencias. *Conferencia acerca de las pruebas de software*. 2014.
43. Universidad Veracruzana. *Universidad Veracruzana*. [En línea] 2015. <http://www.uv.mx/personal/jfernandez/files/2010/07/Pruebas-de-Integracion.pdf>.
44. Martínez, Ing Eduardo Salazar. Portal de e-gobierno, inclusão digital e sociedade do conhecimento. *Portal de e-gobierno, inclusão digital e sociedade do conhecimento*. [En línea] 2015. [Citado el: 25 de mayo de 2015.] <http://www.egov.ufsc.br/portal/conteudo/propuesta-de-procedimiento-para-realizar-pruebas-de-caja-blanca-las-aplicaciones-que-se-des>.
45. Beck, Kent. *XP Explained*. s.l. : Erich Gamma, 2004.
46. Moss, Adrian Holovaty y Jacob Kaplan. *Django Book*. s.l. : Jeremy Dunck, 2008.
47. Ing. Karel Rodríguez Carmenates, Ing. Félix González Martínez. Fordes. *Fordes*. [En línea] noviembre de 2014. [Citado el: 30 de noviembre de 2014.] <http://semanatecnologica.fordes.co.cu/?q=node/1552>.
48. GeoProfile. *GeoProfile*. [En línea] 2011. http://www.dpi.ufv.br/projetos/geoprofile/tutoriais/Visual_Paradigm_for_UML_Tutorial_english.pdf.

49. García, Joaquín. IngenieroSoftware. *IngenieroSoftware*. [En línea] 7 de Mayo de 2005. [Citado el: 20 de Enero de 2015.] <http://www.ingenierosoftware.com/analisisydiseno/uml.php>.
50. Patricio Letelier, María del Carmen Panadés. *Metodologías Ágiles para el desarrollo de Software: Extreme Programing (XP)*. Departamento de Sistemas Informáticos y Computación, Universidad Politécnica de Valencia. Valencia : s.n.
51. Duque, Raúl González. *Python para todos*. España : Creative Commons Reconocimiento.
52. Python Software Foundation. *Python Software Foundation*. [En línea] 1990-2015. [Citado el: 16 de noviembre de 2014.] <https://docs.python.org/3/tutorial/index.html>.
53. Pincirolí, Lic. Fernando. SEDICI. *SEDICI*. [En línea] 2003-2015. [Citado el: 2 de febrero de 2015.] <http://sedici.unlp.edu.ar/handle/10915/23050>.
54. Daniel López Ortega, Jessica Andrés Santa Villa. Universidad tecnológica de Pereira. *Universidad tecnológica de Pereira*. [En línea] 2015. [Citado el: 15 de Enero de 2015.] <http://repositorio.utp.edu.co/dspace/bitstream/11059/2737/1/0053L864e.pdf>.

UCI: Universidad de las Ciencias Informáticas.

TLM: Centro de Telemática.

SGIR: Sistema Gestores de Inventarios de Red.

GRHS: Gestor de recursos de hardware y software

OCS Inventory NG: Open Computer System Inventory Next Generation (Inventario de Sistema de Computadora abierto a la Próxima Generación).

BD: Base de datos.

XP: Extreme Programming.

CCS: Hojas de estilo en cascada.

HTML: Lenguaje de Marcación de Hipertexto.

SGBD: Sistema Gestor de Bases de Datos.

CASE: Computer Aided Software Engineering (Computadora de ayuda a la ingeniería de software).

JSON: JavaScript Object Notation (Notación de objetos javascript).

Plugins: Software que se relaciona con otro para aportarle una función nueva.

HU: Historia de usuario

TI: Tareas de ingeniería

Tarjetas CRC: Tarjetas Clase-Responsabilidad-Colaboración.

MTV: Model-Template-View (Modelo-Plantilla-Vista).

GRASP: General Responsibility Assignment Software Patterns (Patrones de Software para la asignación de Responsabilidad).

ORM: Mapeador de objetos relacional.

API: Interfaz de programación de aplicaciones.

Anexos

Anexo I: Transformación y migración de base de datos

La tabla Hardware la cual cuenta con 33 columnas al migrar su información se establece en las tablas Agent, Computer, SystemUser, OperatingSystem y Microprocessor; ya que esta tabla Hardware contiene toda esta información y la BD del GRHS no la trata de la misma manera. Además la tabla Bios de la BD del OCS Inventory se migrar hacia una tabla con el mismo nombre en GRHS, aunque no coinciden en el número de columnas. También se migra la información referente a los controllers del OCS hacia la llamada Controller en GRHS. La tabla NetworkInterface de GRHS recibe a la información que contiene la tabla networks del OCS Inventory. Asimismo sucede con la tabla Program que acoge la información contenida en software. De igual forma la tabla RAM recibe los datos guardados en la tabla memories. Del mismo modo la tabla monitors de OCS Inventory migra la información a la tabla Monitor. Otro caso particular resulta la tabla inputs ya que al migrar los datos estos ocupan un espacio en las tablas Keyboard y Mouse. La tabla storages le cede sus datos a la tabla CDROM. Por último la tabla Printer recibe la información de printers. Los elementos se simbolizan mediante números, es decir los números establecen de y hacia donde se migra la información.

Tabla 12: Transformación y migración de la base de datos de OCS Inventory NG a GRHS

OCS	GRHS												
<table border="1"> <tr><td>Hardware</td></tr> <tr><td>ID</td></tr> <tr><td>DEVICEID</td></tr> <tr><td>1-NAME</td></tr> <tr><td>2-WORKGROUP</td></tr> <tr><td>USERDOMAIN</td></tr> <tr><td>4-OSNAME</td></tr> </table>	Hardware	ID	DEVICEID	1-NAME	2-WORKGROUP	USERDOMAIN	4-OSNAME	<table border="1"> <tr><td>Agent</td></tr> <tr><td>id_agent</td></tr> <tr><td>1-host</td></tr> <tr><td>1-host_migration</td></tr> <tr><td>Computer</td></tr> </table>	Agent	id_agent	1-host	1-host_migration	Computer
Hardware													
ID													
DEVICEID													
1-NAME													
2-WORKGROUP													
USERDOMAIN													
4-OSNAME													
Agent													
id_agent													
1-host													
1-host_migration													
Computer													

5-OSVERSION	2-Domain
OSCOMMENTS	1-Hostname
7-PROCESSORT	
8-PROCESSORS	SystemUser
PROCESSORN	is_local
MEMORY	3-Name
SWAP	is_logged
IPADDR	computer = Computer
DNS	
DEFAULTGATEWAY	OperatingSystem
ETIME	5-version
LASTDATE	4-name
LASTCOME	6-uuid
QUALITY	
FIDELITY	Microprocessor
3-USERID	serial_number
`TYPE`	7-Versión
DESCRIPTION	Voltaje
WINCOMPANY	8-max_speed

<table border="1"> <tr><td>WINOWNER</td></tr> <tr><td>WINPRODID</td></tr> <tr><td>WINPRODKEY</td></tr> <tr><td>USERAGENT</td></tr> <tr><td>CHECKSUM</td></tr> <tr><td>SSTATE</td></tr> <tr><td>IPSRC</td></tr> <tr><td>6-UUID</td></tr> </table>	WINOWNER	WINPRODID	WINPRODKEY	USERAGENT	CHECKSUM	SSTATE	IPSRC	6-UUID	<table border="1"> <tr><td>Vendor</td></tr> <tr><td>family</td></tr> </table>	Vendor	family				
WINOWNER															
WINPRODID															
WINPRODKEY															
USERAGENT															
CHECKSUM															
SSTATE															
IPSRC															
6-UUID															
Vendor															
family															
<table border="1"> <tr><td>Bios</td></tr> <tr><td>HARDWARE_ID</td></tr> <tr><td>9-SMANUFACTURER</td></tr> <tr><td>SMODEL</td></tr> <tr><td>SSN</td></tr> <tr><td>`TYPE`</td></tr> <tr><td>BMANUFACTURER</td></tr> <tr><td>10-BVERSION</td></tr> <tr><td>11-BDATE</td></tr> <tr><td>ASSETTAG</td></tr> </table>	Bios	HARDWARE_ID	9-SMANUFACTURER	SMODEL	SSN	`TYPE`	BMANUFACTURER	10-BVERSION	11-BDATE	ASSETTAG	<table border="1"> <tr><td>Bios</td></tr> <tr><td>9-manufacturer</td></tr> <tr><td>10-version</td></tr> <tr><td>11-release_date</td></tr> </table>	Bios	9-manufacturer	10-version	11-release_date
Bios															
HARDWARE_ID															
9-SMANUFACTURER															
SMODEL															
SSN															
`TYPE`															
BMANUFACTURER															
10-BVERSION															
11-BDATE															
ASSETTAG															
Bios															
9-manufacturer															
10-version															
11-release_date															

<table border="1"> <thead> <tr> <th>Controllers</th> </tr> </thead> <tbody> <tr> <td>ID</td> </tr> <tr> <td>HARDWARE_ID</td> </tr> <tr> <td>12-MANUFACTURER</td> </tr> <tr> <td>13-NAME</td> </tr> <tr> <td>CAPTION</td> </tr> <tr> <td>DESCRIPTION</td> </tr> <tr> <td>VERSION</td> </tr> <tr> <td>`TYPE`</td> </tr> </tbody> </table>	Controllers	ID	HARDWARE_ID	12-MANUFACTURER	13-NAME	CAPTION	DESCRIPTION	VERSION	`TYPE`	<table border="1"> <thead> <tr> <th>Controller</th> </tr> </thead> <tbody> <tr> <td>12-manufacturer</td> </tr> <tr> <td>device</td> </tr> <tr> <td>13-name</td> </tr> </tbody> </table>	Controller	12-manufacturer	device	13-name		
Controllers																
ID																
HARDWARE_ID																
12-MANUFACTURER																
13-NAME																
CAPTION																
DESCRIPTION																
VERSION																
`TYPE`																
Controller																
12-manufacturer																
device																
13-name																
<table border="1"> <thead> <tr> <th>networks</th> </tr> </thead> <tbody> <tr> <td>ID</td> </tr> <tr> <td>HARDWARE_ID</td> </tr> <tr> <td>14-DESCRIPTION</td> </tr> <tr> <td>`TYPE`</td> </tr> <tr> <td>TYPEMIB</td> </tr> <tr> <td>SPEED</td> </tr> <tr> <td>15-MACADDR</td> </tr> </tbody> </table>	networks	ID	HARDWARE_ID	14-DESCRIPTION	`TYPE`	TYPEMIB	SPEED	15-MACADDR	<table border="1"> <thead> <tr> <th>NetworkInterface</th> </tr> </thead> <tbody> <tr> <td>14-name</td> </tr> <tr> <td>16-ip_address</td> </tr> <tr> <td>18-gateway</td> </tr> <tr> <td>15-mac</td> </tr> <tr> <td>17-network_mask</td> </tr> <tr> <td>computer = Computer</td> </tr> </tbody> </table>	NetworkInterface	14-name	16-ip_address	18-gateway	15-mac	17-network_mask	computer = Computer
networks																
ID																
HARDWARE_ID																
14-DESCRIPTION																
`TYPE`																
TYPEMIB																
SPEED																
15-MACADDR																
NetworkInterface																
14-name																
16-ip_address																
18-gateway																
15-mac																
17-network_mask																
computer = Computer																

<table border="1"> <tr><td>`STATUS`</td></tr> <tr><td>16-IPADDRESS</td></tr> <tr><td>17-IPMASK</td></tr> <tr><td>18-IPGATEWAY</td></tr> <tr><td>IPSUBNET</td></tr> <tr><td>IPDHCP</td></tr> <tr><td>VIRTUALDEV</td></tr> </table>	`STATUS`	16-IPADDRESS	17-IPMASK	18-IPGATEWAY	IPSUBNET	IPDHCP	VIRTUALDEV											
`STATUS`																		
16-IPADDRESS																		
17-IPMASK																		
18-IPGATEWAY																		
IPSUBNET																		
IPDHCP																		
VIRTUALDEV																		
<table border="1"> <tr><td>softwares</td></tr> <tr><td>ID</td></tr> <tr><td>HARDWARE_ID</td></tr> <tr><td>PUBLISHER</td></tr> <tr><td>19-NAME</td></tr> <tr><td>20-VERSION</td></tr> <tr><td>FOLDER</td></tr> <tr><td>COMMENTS</td></tr> <tr><td>FILENAME</td></tr> <tr><td>21-FILESIZE</td></tr> <tr><td>SOURCE</td></tr> </table>	softwares	ID	HARDWARE_ID	PUBLISHER	19-NAME	20-VERSION	FOLDER	COMMENTS	FILENAME	21-FILESIZE	SOURCE	<table border="1"> <tr><td>Program</td></tr> <tr><td>21-size</td></tr> <tr><td>20-version</td></tr> <tr><td>manufacturer</td></tr> <tr><td>22-installation_date</td></tr> <tr><td>19-name</td></tr> </table>	Program	21-size	20-version	manufacturer	22-installation_date	19-name
softwares																		
ID																		
HARDWARE_ID																		
PUBLISHER																		
19-NAME																		
20-VERSION																		
FOLDER																		
COMMENTS																		
FILENAME																		
21-FILESIZE																		
SOURCE																		
Program																		
21-size																		
20-version																		
manufacturer																		
22-installation_date																		
19-name																		

<table border="1"> <tr> <td data-bbox="315 212 696 302"> <p>22-INSTALLDATE</p> </td> </tr> </table>	<p>22-INSTALLDATE</p>																			
<p>22-INSTALLDATE</p>																				
<table border="1"> <tr> <th data-bbox="315 522 696 611">Memories</th> </tr> <tr> <td data-bbox="315 611 696 699">ID</td> </tr> <tr> <td data-bbox="315 699 696 787">HARDWARE_ID</td> </tr> <tr> <td data-bbox="315 787 696 875">23-CAPTION</td> </tr> <tr> <td data-bbox="315 875 696 963">DESCRIPTION</td> </tr> <tr> <td data-bbox="315 963 696 1052">24-CAPACITY</td> </tr> <tr> <td data-bbox="315 1052 696 1140">PURPOSE</td> </tr> <tr> <td data-bbox="315 1140 696 1228">25-`TYPE`</td> </tr> <tr> <td data-bbox="315 1228 696 1316">26-SPEED</td> </tr> <tr> <td data-bbox="315 1316 696 1404">NUMSLOTS</td> </tr> <tr> <td data-bbox="315 1404 696 1493">27-SERIALNUMBER</td> </tr> </table>	Memories	ID	HARDWARE_ID	23-CAPTION	DESCRIPTION	24-CAPACITY	PURPOSE	25-`TYPE`	26-SPEED	NUMSLOTS	27-SERIALNUMBER	<table border="1"> <tr> <th data-bbox="902 522 1245 611">RAM</th> </tr> <tr> <td data-bbox="902 611 1245 699">25- typeRAM</td> </tr> <tr> <td data-bbox="902 699 1245 787">24-size</td> </tr> <tr> <td data-bbox="902 787 1245 875">26-speed</td> </tr> <tr> <td data-bbox="902 875 1245 963">23-locator</td> </tr> <tr> <td data-bbox="902 963 1245 1052">Manufacturer</td> </tr> <tr> <td data-bbox="902 1052 1245 1140">data_with</td> </tr> <tr> <td data-bbox="902 1140 1245 1228">27-serial_number</td> </tr> </table>	RAM	25- typeRAM	24-size	26-speed	23-locator	Manufacturer	data_with	27-serial_number
Memories																				
ID																				
HARDWARE_ID																				
23-CAPTION																				
DESCRIPTION																				
24-CAPACITY																				
PURPOSE																				
25-`TYPE`																				
26-SPEED																				
NUMSLOTS																				
27-SERIALNUMBER																				
RAM																				
25- typeRAM																				
24-size																				
26-speed																				
23-locator																				
Manufacturer																				
data_with																				
27-serial_number																				
<table border="1"> <tr> <th data-bbox="315 1499 696 1587">Monitors</th> </tr> <tr> <td data-bbox="315 1587 696 1675">ID</td> </tr> <tr> <td data-bbox="315 1675 696 1764">HARDWARE_ID</td> </tr> <tr> <td data-bbox="315 1764 696 1852">28-MANUFACTURER</td> </tr> </table>	Monitors	ID	HARDWARE_ID	28-MANUFACTURER	<table border="1"> <tr> <th data-bbox="902 1499 1245 1587">Monitor</th> </tr> <tr> <td data-bbox="902 1587 1245 1675">28-manufacturer</td> </tr> <tr> <td data-bbox="902 1675 1245 1764">model</td> </tr> <tr> <td data-bbox="902 1764 1245 1852">29-serial_number</td> </tr> </table>	Monitor	28-manufacturer	model	29-serial_number											
Monitors																				
ID																				
HARDWARE_ID																				
28-MANUFACTURER																				
Monitor																				
28-manufacturer																				
model																				
29-serial_number																				

<table border="1"> <tr><td>CAPTION</td></tr> <tr><td>DESCRIPTION</td></tr> <tr><td>`TYPE`</td></tr> <tr><td>29-SERIAL</td></tr> </table>	CAPTION	DESCRIPTION	`TYPE`	29-SERIAL																
CAPTION																				
DESCRIPTION																				
`TYPE`																				
29-SERIAL																				
<table border="1"> <tr><td>inputs</td></tr> <tr><td>ID</td></tr> <tr><td>HARDWARE_ID</td></tr> <tr><td>30-`TYPE`</td></tr> <tr><td>MANUFACTURER</td></tr> <tr><td>31-CAPTION</td></tr> <tr><td>32-DESCRIPTION</td></tr> <tr><td>33-INTERFACE</td></tr> <tr><td>POINTTYPE</td></tr> </table>	inputs	ID	HARDWARE_ID	30-`TYPE`	MANUFACTURER	31-CAPTION	32-DESCRIPTION	33-INTERFACE	POINTTYPE	<table border="1"> <tr><td>Keyboard</td></tr> <tr><td>31 y 32-id_family</td></tr> <tr><td>vendor</td></tr> <tr><td>id_product</td></tr> <tr><td>33-interface</td></tr> <tr><td>Mouse</td></tr> <tr><td>31 y 32-id_family</td></tr> <tr><td>vendor</td></tr> <tr><td>id_product</td></tr> <tr><td>33-interface</td></tr> </table>	Keyboard	31 y 32-id_family	vendor	id_product	33-interface	Mouse	31 y 32-id_family	vendor	id_product	33-interface
inputs																				
ID																				
HARDWARE_ID																				
30-`TYPE`																				
MANUFACTURER																				
31-CAPTION																				
32-DESCRIPTION																				
33-INTERFACE																				
POINTTYPE																				
Keyboard																				
31 y 32-id_family																				
vendor																				
id_product																				
33-interface																				
Mouse																				
31 y 32-id_family																				
vendor																				
id_product																				
33-interface																				
<table border="1"> <tr><td>storages</td></tr> <tr><td>ID</td></tr> <tr><td>HARDWARE_ID</td></tr> <tr><td>MANUFACTURER</td></tr> </table>	storages	ID	HARDWARE_ID	MANUFACTURER	<table border="1"> <tr><td>CDROM</td></tr> <tr><td>37- serial_number</td></tr> <tr><td>interface</td></tr> <tr><td>model</td></tr> </table>	CDROM	37- serial_number	interface	model											
storages																				
ID																				
HARDWARE_ID																				
MANUFACTURER																				
CDROM																				
37- serial_number																				
interface																				
model																				

<table border="1"> <tr><td>34- NAME (</td></tr> <tr><td>MODEL</td></tr> <tr><td>35- DESCRIPTION</td></tr> <tr><td>36- `TYPE`</td></tr> <tr><td>DISKSIZE</td></tr> <tr><td>37- SERIALNUMBER</td></tr> <tr><td>FIRMWARE</td></tr> </table>	34- NAME (MODEL	35- DESCRIPTION	36- `TYPE`	DISKSIZE	37- SERIALNUMBER	FIRMWARE	<table border="1"> <tr><td>36- typeCD</td></tr> </table>	36- typeCD		
34- NAME (
MODEL											
35- DESCRIPTION											
36- `TYPE`											
DISKSIZE											
37- SERIALNUMBER											
FIRMWARE											
36- typeCD											
<table border="1"> <tr><td>printers</td></tr> <tr><td>ID</td></tr> <tr><td>HARDWARE_ID</td></tr> <tr><td>38-NAME</td></tr> <tr><td>DRIVER</td></tr> <tr><td>37-PORT</td></tr> </table>	printers	ID	HARDWARE_ID	38-NAME	DRIVER	37-PORT	<table border="1"> <tr><td>Printer</td></tr> <tr><td>38-vendor</td></tr> <tr><td>Id_product</td></tr> <tr><td>37-interface</td></tr> </table>	Printer	38-vendor	Id_product	37-interface
printers											
ID											
HARDWARE_ID											
38-NAME											
DRIVER											
37-PORT											
Printer											
38-vendor											
Id_product											
37-interface											

Anexo II: Muestras de historias de usuario

Tabla 13: HU #2 Gestionar conexión al SGBD MySQL

Historia de Usuario	
Número: 1	Usuario: Usuario
Nombre historia: Gestionar conexión al SGBD MySQL	

Modificación de Historia de Usuario: Una

Prioridad en negocio: Alta

Riesgo en desarrollo: Alto

Puntos estimados: 2

Iteración asignada: 2

Programador responsable: Bladimir Almeida Rodríguez y Ronal Ledesma López

Descripción: El usuario selecciona dentro de la opción Conexiones, MySQL y se muestra una ventana brindando las opciones de ingresar, modificar y eliminar conexión. Los datos que se deben introducir son:

- Usuario SGBD
- Nombre BD
- Localización BD
- Puerto BD

Observaciones: Si no se introducen todos los datos requeridos no se guardan los datos de la posible conexión.

Si los datos entrados coinciden con datos registrados, no se guardarán los datos de esa conexión.

Prototipo de Interfaz

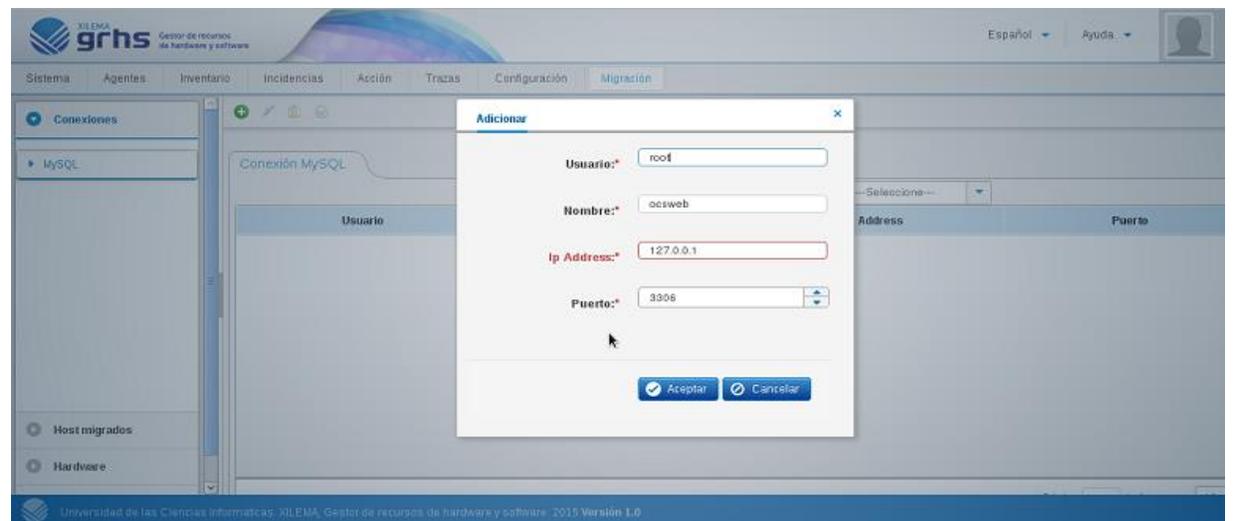


Tabla 14: HU # 2 Establecer conexión con el gestor de base de datos MySQL

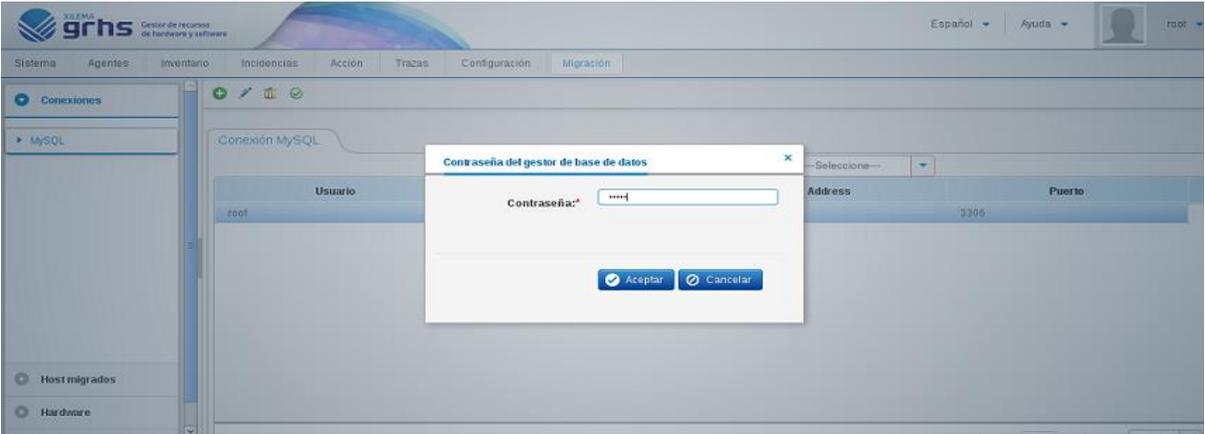
Historias de Usuario	
Número: 2	Usuario: Administrador
Nombre historia: Establecer conexión con el gestor de base de datos MySQL	
Modificación de historia de usuario: Una	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Ronal Ledesma López y Bladimir Almeida Rodríguez	
<p>Descripción: El usuario selecciona una de las posibles conexiones registradas y presiona la botón conectar, luego se muestra una ventana para que sea introducida la contraseña de acceso al SGBD MySQL, se presiona el botón aceptar y se establece la conexión al gestor para esa base de datos seleccionada.</p>	
<p>Observaciones: Si se introduce una contraseña incorrecta, no se establecerá la conexión a la base de datos.</p>	
<p>Prototipo de Interfaz :</p> 	

Tabla 15: HU # 4 Mostrar información migrada de OCS Inventory NG a GRHS

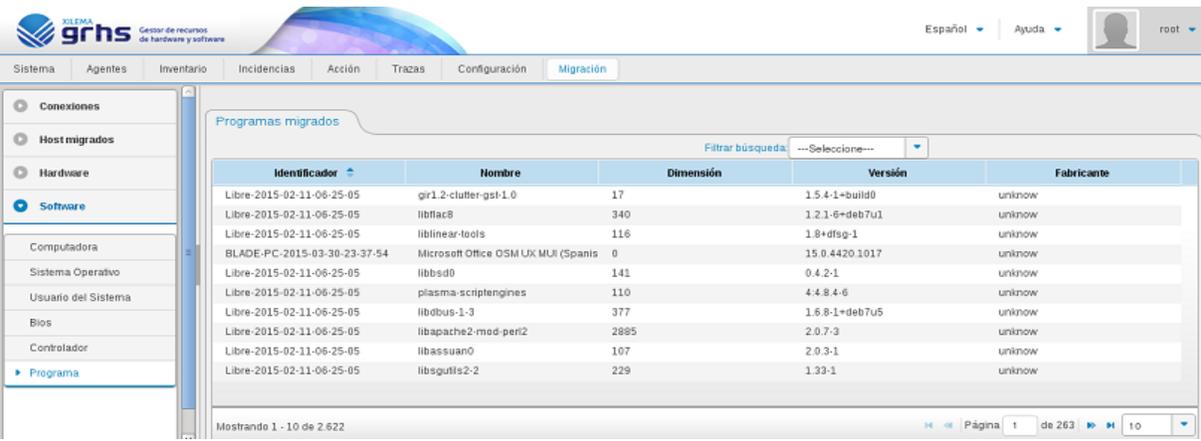
Historias de Usuario																																																								
Número: 4	Usuario: Usuario																																																							
Nombre historia: Mostrar información migrada de OCS Inventory NG a GRHS																																																								
Modificación de historia de usuario: Ninguna																																																								
Prioridad en negocio: Media	Riesgo en desarrollo: Medio																																																							
Puntos estimados: 2	Iteración asignada: 3																																																							
Programador responsable: Ronal Ledesma López																																																								
Descripción: El usuario accede a la información migrada, seleccionando el tipo de información que desea consultar.																																																								
Observaciones: Si la información no se puede consultar, debe esperar a que se termine la migración, pues no ha sido migrada esa información aún.																																																								
Prototipo de Interfaz:																																																								
 <p>The screenshot shows the GRHS web application interface. At the top, there is a navigation menu with options: Sistema, Agentes, Inventario, Incidencias, Acción, Trazas, Configuración, and Migración. The 'Migración' tab is active. On the left, there is a sidebar with a tree view containing: Conexiones, Host migrados, Hardware, Software (selected), Computadora, Sistema Operativo, Usuario del Sistema, Bios, Controlador, and Programa. The main content area displays a table titled 'Programas migrados' with a search filter. The table contains the following data:</p> <table border="1"> <thead> <tr> <th>Identificador</th> <th>Nombre</th> <th>Dimensión</th> <th>Versión</th> <th>Fabricante</th> </tr> </thead> <tbody> <tr> <td>Libre-2015-02-11-06-25-05</td> <td>gir1.2-clutter-gst-1.0</td> <td>17</td> <td>1.5.4-1+build0</td> <td>unknow</td> </tr> <tr> <td>Libre-2015-02-11-06-25-05</td> <td>libflac8</td> <td>340</td> <td>1.2.1-6+deb7u1</td> <td>unknow</td> </tr> <tr> <td>Libre-2015-02-11-06-25-05</td> <td>liblinear-tools</td> <td>116</td> <td>1.8+dfsg-1</td> <td>unknow</td> </tr> <tr> <td>BLADE-PC-2015-03-30-23-37-54</td> <td>Microsoft Office DSM UX MUI (Spanis</td> <td>0</td> <td>15.0.4420.1017</td> <td>unknow</td> </tr> <tr> <td>Libre-2015-02-11-06-25-05</td> <td>libbsd0</td> <td>141</td> <td>0.4.2-1</td> <td>unknow</td> </tr> <tr> <td>Libre-2015-02-11-06-25-05</td> <td>plasma-scriptengines</td> <td>110</td> <td>4.4.8-4-6</td> <td>unknow</td> </tr> <tr> <td>Libre-2015-02-11-06-25-05</td> <td>libdbus-1-3</td> <td>377</td> <td>1.6.8-1+deb7u5</td> <td>unknow</td> </tr> <tr> <td>Libre-2015-02-11-06-25-05</td> <td>libapache2-mod-perl2</td> <td>2885</td> <td>2.0.7-3</td> <td>unknow</td> </tr> <tr> <td>Libre-2015-02-11-06-25-05</td> <td>libassuan0</td> <td>107</td> <td>2.0.3-1</td> <td>unknow</td> </tr> <tr> <td>Libre-2015-02-11-06-25-05</td> <td>libsgutils2-2</td> <td>229</td> <td>1.33-1</td> <td>unknow</td> </tr> </tbody> </table> <p>At the bottom of the table, it says 'Mostrando 1 - 10 de 2.622'. The footer of the interface shows 'Página 1 de 263' and '10'.</p>		Identificador	Nombre	Dimensión	Versión	Fabricante	Libre-2015-02-11-06-25-05	gir1.2-clutter-gst-1.0	17	1.5.4-1+build0	unknow	Libre-2015-02-11-06-25-05	libflac8	340	1.2.1-6+deb7u1	unknow	Libre-2015-02-11-06-25-05	liblinear-tools	116	1.8+dfsg-1	unknow	BLADE-PC-2015-03-30-23-37-54	Microsoft Office DSM UX MUI (Spanis	0	15.0.4420.1017	unknow	Libre-2015-02-11-06-25-05	libbsd0	141	0.4.2-1	unknow	Libre-2015-02-11-06-25-05	plasma-scriptengines	110	4.4.8-4-6	unknow	Libre-2015-02-11-06-25-05	libdbus-1-3	377	1.6.8-1+deb7u5	unknow	Libre-2015-02-11-06-25-05	libapache2-mod-perl2	2885	2.0.7-3	unknow	Libre-2015-02-11-06-25-05	libassuan0	107	2.0.3-1	unknow	Libre-2015-02-11-06-25-05	libsgutils2-2	229	1.33-1	unknow
Identificador	Nombre	Dimensión	Versión	Fabricante																																																				
Libre-2015-02-11-06-25-05	gir1.2-clutter-gst-1.0	17	1.5.4-1+build0	unknow																																																				
Libre-2015-02-11-06-25-05	libflac8	340	1.2.1-6+deb7u1	unknow																																																				
Libre-2015-02-11-06-25-05	liblinear-tools	116	1.8+dfsg-1	unknow																																																				
BLADE-PC-2015-03-30-23-37-54	Microsoft Office DSM UX MUI (Spanis	0	15.0.4420.1017	unknow																																																				
Libre-2015-02-11-06-25-05	libbsd0	141	0.4.2-1	unknow																																																				
Libre-2015-02-11-06-25-05	plasma-scriptengines	110	4.4.8-4-6	unknow																																																				
Libre-2015-02-11-06-25-05	libdbus-1-3	377	1.6.8-1+deb7u5	unknow																																																				
Libre-2015-02-11-06-25-05	libapache2-mod-perl2	2885	2.0.7-3	unknow																																																				
Libre-2015-02-11-06-25-05	libassuan0	107	2.0.3-1	unknow																																																				
Libre-2015-02-11-06-25-05	libsgutils2-2	229	1.33-1	unknow																																																				

Tabla 16: HU # 5 Editar el nombre de los host migrados

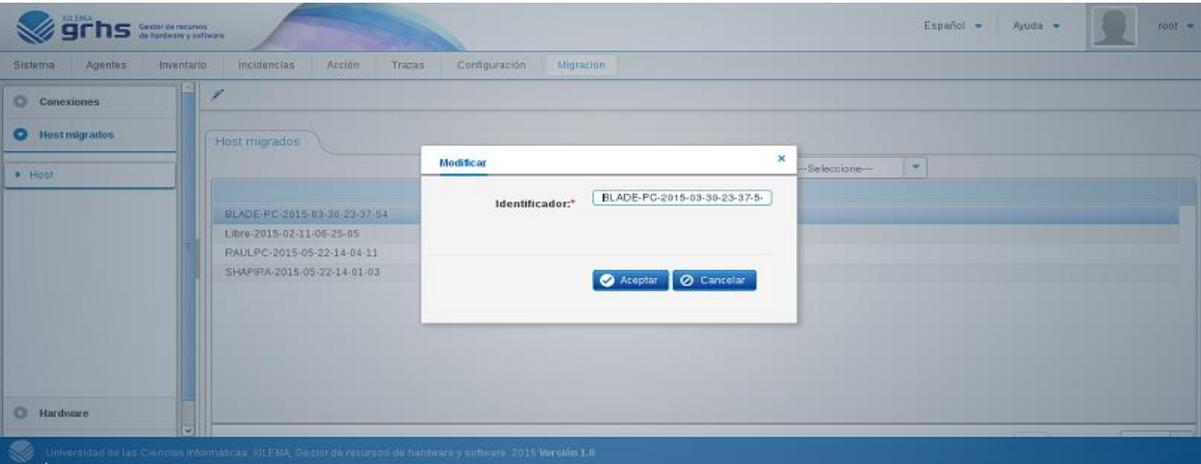
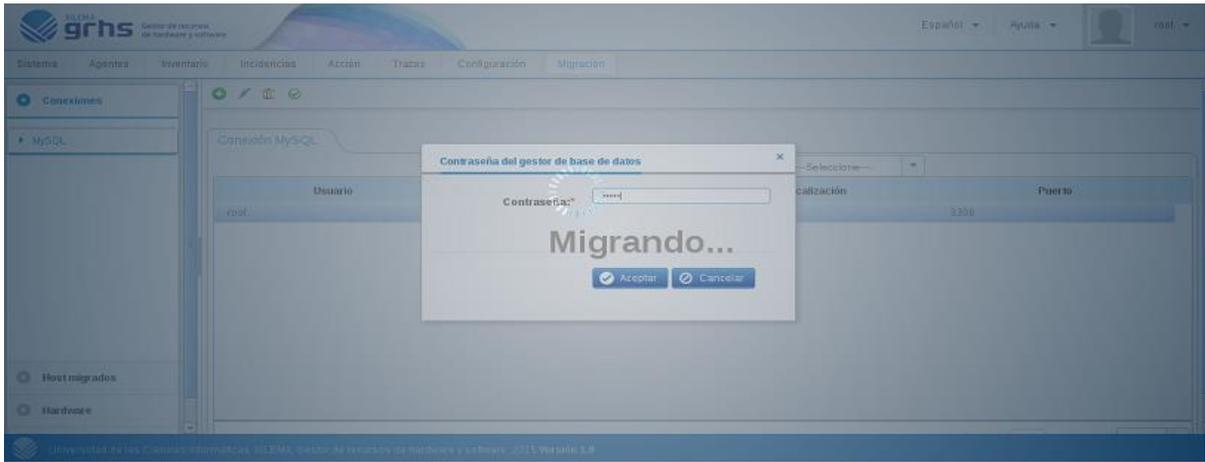
Historias de Usuario	
Número: 5	Usuario: Administrador
Nombre historia: Editar el nombre de los host migrados	
Modificación de historia de usuario: Ninguna	
Prioridad en negocio: Media	Riesgo en desarrollo: Medio
Puntos estimados: 1	Iteración asignada: 2
Programador responsable: Bladimir Almeida Rodríguez	
<p>Descripción: Se selecciona la funcionalidad host migrados, y se muestra una listado de estos, luego se selecciona el host, se presiona el botón editar y se le introduce el nuevo nombre que desee el usuario.</p>	
<p>Observaciones: Es necesario que se haya realizado la migración.</p> <p>No se debe nombrar dos o varios host con el mismo nombre.</p>	
<p>Prototipo de Interfaz:</p> 	

Tabla 17: HU # 6 Mostrar mensaje acerca de la migración de información

Historias de Usuario	
Número: 6	Usuario: Sistema
Nombre historia: Mostrar mensaje acerca de la migración de información	
Modificación de historia de usuario: Ninguna	
Prioridad en negocio: Baja	Riesgo en desarrollo: Bajo
Puntos estimados: 2	Iteración asignada: 3
Programador responsable: Ronal Ledesma López y Bladimir Almeida Rodríguez	
Descripción: El sistema muestra un mensaje mediante una animación, avisando que se está ejecutando el proceso de migración de la información.	
Observaciones: Es necesario que se haya establecido la conexión.	
Prototipo de Interfaz	
	

Anexo III: Muestras de Tarjetas CRC

Tabla 18: Tarjeta CRC Insert_agent_inventory

Tarjetas CRC	
Clase: Insert_agent_inventory	
Responsabilidades	Colaboraciones
Es el responsable de insertar un agente a la base de datos.	Utils Migration_Agent

Tabla 19: Tarjeta CRC Insert_Software

Tarjetas CRC	
Clase: Insert_Software	
Responsabilidades	Colaboraciones
Es el responsable de insertar la información referente a software en la base de datos.	Utils Bios Controller

Tabla 20: Tarjeta CRC Insert_Hardware

Tarjetas CRC	
Clase: Insert_Hardware	
Responsabilidades	Colaboraciones

<p>Es el responsable de insertar la información referente a hardware en la base de datos.</p>	<p>Utils RAM CDROM</p>
---	--------------------------------

Anexo IV: Muestras de tareas de Ingeniería

Tabla 21: TI # 2 Obtener información de la base de datos de OCS Inventory NG

<p>Tareas de Ingeniería</p>	
<p>Número Tarea: 2</p>	<p>Número Historia de Usuario: 1</p>
<p>Nombre Tarea: Obtener información de la base de datos de OCS Inventory NG</p>	
<p>Tipo de Tarea : Desarrollo</p>	<p>Puntos Estimados:1</p>
<p>Programador Responsable: Bladimir Almeida Rodríguez</p>	
<p>Descripción: El sistema se auxilia del ORM de Django para acceder a los modelos que contienen la información de la base de datos de OCS Inventory NG.</p>	

Tabla 22: TI # 3 Transformar información de la base de datos de OCS Inventory NG

<p>Tareas de Ingeniería</p>	
<p>Número Tarea: 3</p>	<p>Número Historia de Usuario: 1</p>
<p>Nombre Tarea: Transformar información de la base de datos de OCS Inventory NG</p>	
<p>Tipo de Tarea : Desarrollo</p>	<p>Puntos Estimados:0,5</p>
<p>Programador Responsable: Ronal Ledesma López</p>	

Descripción: El sistema transforma los datos que no son asimilados por el SGBD destino PostgreSQL, mediante las Clases Utils, Clean_Software, Clean_Hardware las cuales se encargan de realizarle la transformación de los datos.

Tabla 23: TI # 4 Insertar información en la BD GRHS

Tareas de Ingeniería	
Número Tarea: 4	Número Historia de Usuario: 1
Nombre Tarea: Insertar información en la BD GRHS	
Tipo de Tarea : Desarrollo	Puntos Estimados: 0,5
Programador Responsable: Bladimir Almeida Rodríguez	
Descripción: El sistema emplea funcionalidades encargadas de insertar la información, mediante las clases Insert_agent_inventory, Insert_Hardware e Insert_Software.	

Tabla 24: TI # 6 Editar Conexión a MySQL

Tareas de Ingeniería	
Número Tarea: 6	Número Historia de Usuario: 2
Nombre Tarea: Editar conexión a MySQL	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1
Programador Responsable: Bladimir Almeida Rodríguez	
Descripción: El usuario selecciona una conexión existente, después elige la funcionalidad (Editar), luego se muestra una formulario con los datos de la conexión seleccionada y esta puede ser editada, al enviarlo se hará uso del método DATA.	

Tabla 25: TI # 7 Eliminar Conexión a MySQL

Tareas de Ingeniería	
Número Tarea: 7	Número Historia de Usuario: 2
Nombre Tarea: Eliminar conexión a MySQL	
Tipo de Tarea : Desarrollo	Puntos Estimados:0,5
Programador Responsable: Bladimir Almeida Rodríguez y Ronal Ledesma López	
<p>Descripción: El usuario selecciona una conexión existente, después elige la funcionalidad (Eliminar), luego se muestra una ventana con un mensaje de confirmación para eliminar esa conexión.</p>	

Tabla 26: TI # 8 Establecer conexión con el SGBD MySQL

Tareas de Ingeniería	
Número Tarea: 8	Número Historia de Usuario: 3
Nombre Tarea: Establecer conexión con el GBD MySQL	
Tipo de Tarea : Desarrollo	Puntos Estimados:0,5
Programador Responsable: Ronal Ledesma López y Bladimir Almeida Rodríguez	
<p>Descripción: El usuario selecciona una conexión existente, después elige la funcionalidad (Conectar), y aparece un formulario solicitando la contraseña para establecer la conexión. Si algunos de los parámetros son incorrectos no se establece la conexión, se envía un mensaje de error al usuario notificándole por qué no se realiza la conexión y en caso contrario esta se realiza de manera satisfactoria, acompañada de un mensaje de éxito. Los parámetros son enviados mediante el método DATA.</p>	

Tabla 27: TI # 9 Mostrar información migrada de OCS Inventory NG a GRHS

Tareas de Ingeniería	
Número Tarea: 9	Número Historia de Usuario: 4
Nombre Tarea: Mostrar información migrada de OCS Inventory NG a GRHS	
Tipo de Tarea : Desarrollo	Puntos Estimados: 2
Programador Responsable: Ronal Ledesma López	
Descripción: El usuario selecciona la información que desea consultar, referente a hardware, software o los host migrados, y a raíz de esto aparece un listado con la información correspondiente a la seleccionada. Para mostrar esta información se serializan cada uno de los modelos seleccionados, con el objetivo de convertir esta información en un formato asimilado por la interfaz.	

Tabla 28: TI # 10 Editar el nombre de los host migrados

Tareas de Ingeniería	
Número Tarea: 10	Número Historia de Usuario: 5
Nombre Tarea: Editar el nombre de los host migrados.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1
Programador Responsable: Bladimir Almeida Rodríguez	
Descripción: El usuario selecciona un host de la lista, elige la opción editar, se muestra un formulario con la información referente al nombre, se introduce un nuevo nombre deseado por el usuario y este guarda el cambio realizado. Se hace uso de los serializadores para mostrar el nombre del nuevo host.	

Tabla 29: TI # 11 Mostrar mensaje acerca de la migración de información

Tareas de Ingeniería

Número Tarea: 11	Número Historia de Usuario: 6
Nombre Tarea: Mostrar mensaje acerca de la migración de información.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 2
Programador Responsable: Bladimir Almeida Rodríguez	
Descripción: El sistema hace peticiones Ajax mediante una variable global al servidor, esta variable va a controlar el proceso de migración y mediante una animación anuncia al usuario que se está migrando.	

Anexo V: Muestra de pruebas de aceptación

Tabla 30: Prueba de aceptación # 2 HU Gestionar Conexiones

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observación
El usuario accede al módulo (Migración), y selecciona la opción (Conexiones), luego escoge la conexión y presiona el botón (Modificar), edita los campos deseados y guarda los cambios.		El sistema muestra un mensaje “La operación se realizó con éxito”.	Satisfactorio	

	<p>El usuario accede al módulo migración y selecciona la opción Conexiones, luego escoge la conexión y presiona el botón (Modificar) edita el campo, pero no introduce valores.</p>	<p>El sistema muestra un error "Por favor inserte un valor(es) para este campo(s)"</p>	<p>No se guardan los datos modificados.</p>	
	<p>El usuario accede al módulo (Migración), y selecciona la opción (Conexiones), luego escoge la conexión y presiona el botón (Modificar), edita el campo, pero no introduce la (Localización) correcta.</p>	<p>El sistema muestra un error "Por favor introduzca una dirección IPv4 o IPv6 válida".</p>	<p>No se muestra la información de las conexiones.</p>	

	<p>El usuario accede al módulo (Migración), y selecciona la opción (Conexiones), pero el sistema no tiene conexión con el servidor.</p>	<p>El sistema no responde.</p>	<p>No se muestra la información de las conexiones.</p>	
--	---	--------------------------------	--	--

Tabla 31: Prueba de aceptación # 3 HU Gestionar Conexiones

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observación
<p>El usuario accede al módulo (Migración), y selecciona la opción (Conexiones), luego escoge la conexión y presiona el botón (Eliminar) y confirma la eliminación, siendo eliminados los datos.</p>		<p>El sistema muestra un mensaje “La operación se realizó con éxito”.</p>	<p>Satisfactorio</p>	

	<p>El usuario accede al módulo (Migración), y selecciona la opción (Conexiones), pero el sistema no tiene conexión con el servidor.</p>	<p>El sistema no responde.</p>	<p>No se muestra la información de las conexiones.</p>	
--	---	--------------------------------	--	--

Tabla 32: Pruebas de aceptación # 4 HU Establecer conexión con el gestor de base de datos MySQL

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observación
<p>El usuario accede al módulo (Migración), y selecciona la opción (Conexiones), luego escoge la conexión y presiona el botón (Conectar) e introduce la contraseña para conectarse con el SGBD MySQL y presiona el</p>		<p>El sistema muestra el mensaje "Migrando..."</p>	<p>Satisfactorio</p>	<p>El sistema bloqueará la interfaz hasta que termine el proceso de migración.</p>

<p>botón (Conectar).</p>				
	<p>El usuario accede al módulo (Migración), y selecciona la opción (Conexiones), luego escoge la conexión y presiona el botón (Conectar) y no introduce los datos requeridos por ese campo.</p>	<p>El sistema muestra un error "Por favor inserte un valor para este campo."</p>	<p>No se establece la conexión.</p>	
	<p>El usuario accede al módulo (Migración), y selecciona la opción (Conexiones), luego escoge la conexión, introduce la contraseña incorrecta del SGBD MySQL y presiona el botón (Conectar).</p>	<p>El sistema muestra un error "No se ha establecido la conexión. Error: "Respuesta del SGBD MySQL"</p>	<p>No se establece la conexión.</p>	

Tabla 33: Prueba de aceptación # 5 HU Mostrar información migrada de OCS Inventory NG a GRHS

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observación
El usuario accede al módulo (Migración) y selecciona la opción (Host Migrados).		El sistema muestra la información referente a los host migrados durante el proceso de migración.	Satisfactorio	
	El usuario accede al módulo (Migración) y selecciona la opción (Host Migrados), pero el sistema no tiene conexión con el servidor.	El sistema no responde.	No se muestra la información referente a los host migrados.	

Tabla 34: Prueba de aceptación # 6 HU Mostrar información migrada de OCS Inventory NG a GRHS

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observación
El usuario accede al módulo (Migración) y selecciona la opción (Hardware).		El sistema muestra la información referente al hardware migrado.	Satisfactorio	
	El usuario accede al módulo (Migración) y selecciona la opción (Hardware), pero el sistema no tiene conexión con el servidor.	El sistema no responde.	No se muestra la información referente al Hardware migrado.	

Tabla 35: Prueba de aceptación # 7 HU Mostrar información migrada de OCS Inventory NG a GRHS

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observación
El usuario accede al módulo (Migración) y selecciona la opción (Software).		El sistema muestra la información referente al software migrado.	Satisfactorio	
	El usuario accede al módulo (Migración) y selecciona la opción (Software), pero el sistema no tiene conexión con el servidor.	El sistema no responde.	No se muestra la información referente al Software migrado.	

Tabla 36: Prueba de aceptación # 8 HU Editar el nombre de los host migrados

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observación
El usuario accede al módulo (Migración) y selecciona la opción (Host Migrados) y pulsa el botón (Editar) y le cambia el nombre al host seleccionado.		El sistema muestra un mensaje "La operación se realizó con éxito".	Satisfactorio	
	El usuario accede al módulo (Migración) y selecciona la opción (Host Migrados) y pulsa el botón (Editar) y deja el campo nombre del host en blanco.	El sistema muestra un error "Por favor inserte un valor para este campo."	No se guarda la información modificada.	
	El usuario accede al módulo (Migración) y selecciona la opción (Host Migrados) y pulsa el botón	El sistema no responde.	No se modifica la información.	

	(Editar), pero el sistema no tiene conexión con el servidor			
--	---	--	--	--