



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

***Desarrollo de un Sistema Informático de Apoyo a la Migración a
Software Libre***

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

***Autor:
Carlos Javier Palacios Morales.***

***Tutor:
Ing. William Feal Delgado.***

Agradecimientos .

Quisiera agradecer a todos los que de una forma u otra han contribuido con su ayuda y sin los cuales no hubiese sido posible la realización de este trabajo, en especial:

- ☞ A mi Dios y Salvador Jesucristo por ser mi todo en todo.*
- ☞ A mi Madre y mi Padre, por su inmenso sacrificio, su dedicación y su amor.*
- ☞ A mi Familia que siempre me ha respaldado y apoyado.*
- ☞ A mis Pastores Jorge, Marisel, Roberto y Magalis por su apoyo y continua sabiduría.*
- ☞ A mis Hermanos, quienes han estado conmigo siempre.*
- ☞ A mis Amigos, quienes se han mostrado amigos en todo tiempo.*
- ☞ A mi Tutor, William Feal, por su paciencia y confianza.*

“A Todos Muchas Gracias”

Al Gran Yo Soy

*“No digas a tu prójimo: Anda, y vuelve,
Y mañana te daré, Cuando tienes contigo qué darle.”*

Proverbios 3:18, RV60

Resumen

El software libre es un asunto de actualidad y promete imponerse gradualmente en la industria informática como una herramienta de evolución continua. Además de presentar una enorme oportunidad para la investigación y la innovación es también una perfecta oportunidad para educar y formar valores necesarios en la sociedad actual. Dadas las probadas ventajas del software libre, el MES (Ministerio de la Educación Superior) se proyecta en aras de difundir su utilización en sus dependencias. Es por esto que la Universidad de Cienfuegos trabaja en una estrategia para su implantación. El exclusivo uso de sistemas privativos ha generado una gran dependencia a los mismos por lo que se hace necesario concebir un plan de migración que mediante pasos definidos inserte gradualmente las tecnologías de software libre. El presente trabajo investigativo tiene como objetivo desarrollar un sistema para gestionar la información generada en el proceso de migración a software libre. Con este fin se efectuó una revisión de la bibliografía técnico-especializada y del estado del arte en la materia, a fin de determinar las tecnologías y métodos a emplear para desarrollar e implementar el sistema. Además se utilizó el lenguaje de modelado UML para el análisis, diseño e implementación de la solución propuesta, siguiendo lo establecido por el Proceso Unificado de Desarrollo de Software (RUP).

ÍNDICE

INTRODUCCIÓN	2
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.	8
1.1 INTRODUCCIÓN.	8
1.2 DESCRIPCIÓN DEL DOMINIO DEL PROBLEMA.	8
1.2.1 SOFTWARE LIBRE Y MIGRACIÓN	8
1.2.2 PANORAMA MUNDIAL.....	10
1.2.4 SOFTWARE LIBRE EN LA EDUCACIÓN	11
1.2.4 SOFTWARE LIBRE EN CUBA.....	12
1.2.4 SOFTWARE LIBRE EN LAS UNIVERSIDADES CUBANAS	13
1.3 OBJETO DE ESTUDIO	14
1.3.1 DESCRIPCIÓN GENERAL	14
1.4 SISTEMAS EXISTENTES VINCULADOS AL CAMPO DE ACCIÓN.	16
1.5 DESCRIPCIÓN DEL OBJETO DE INFORMATIZACIÓN.....	19
1.6 TENDENCIAS, METODOLOGÍAS Y/O TECNOLOGÍAS ACTUALES	20
1.6.1 INTRODUCCIÓN	20
1.6.2 FUNDAMENTACIÓN DE LA METODOLOGÍA UTILIZADA	20
1.6.3 TENDENCIAS ACTUALES Y TECNOLOGÍAS A CONSIDERAR	22
1.6.3.1 Arquitectura de Desarrollo de N Capas	22
1.6.3.2 Tecnologías Web	24
1.6.3.3 Sistemas Gestores de Bases de Datos	31
1.6.4 HERRAMIENTAS DE DESARROLLO	32
1.6.6 CONCLUSIONES	36
CAPÍTULO 2: DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA	37
2.1 INTRODUCCIÓN.	37
2.2 MODELO DEL DOMINIO	37
2.2.1 DEFINICIÓN DE LAS ENTIDADES Y LOS CONCEPTOS PRINCIPALES.	37
2.2.2 REGLAS DEL NEGOCIO A CONSIDERAR.....	39
2.2.4 REPRESENTACIÓN DEL MODELO DEL DOMINIO.....	40
2.3 DESCRIPCIÓN DEL SISTEMA PROPUESTO.....	40
EL SISTEMA DE APOYO A LA MIGRACIÓN (SAM) CUMPLE CON LOS SIGUIENTES REQUERIMIENTOS:	42
2.3.1 REQUERIMIENTOS FUNCIONALES.....	42
2.3.3 REQUERIMIENTOS NO FUNCIONALES.....	44
2.4 MODELO DE CASOS DE USO DEL SISTEMA.....	47
2.4.1 ACTORES DEL SISTEMA	47
2.4.2 PAQUETES Y SUS RELACIONES.....	48

2.4.3 DIAGRAMAS DE CASOS DE USO DEL SISTEMA POR PAQUETES.	50
2.4.4 DESCRIPCIÓN DE LOS CASOS DE USO DEL SISTEMA.	52
2.5 CONCLUSIONES	67
CAPÍTULO 3: CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA	68
3.1 INTRODUCCIÓN.	68
3.2 DIAGRAMA DE CLASES DEL DISEÑO.	68
3.3 PRINCIPIOS DE DISEÑO.	92
3.3.1 ESTÁNDARES EN LA INTERFAZ DE LA APLICACIÓN.	92
3.3.2 CONCEPCIÓN GENERAL DE LA AYUDA.	93
3.3.3 TRATAMIENTO DE ERRORES.	93
3.4 DISEÑO DE LA BASE DE DATOS.	94
3.4.1 MODELO LÓGICO DE DATOS.....	95
3.4.2 MODELO FÍSICO DE DATOS.	95
3.5 DIAGRAMA DE IMPLEMENTACIÓN.	96
3.7 ESTUDIO DE FACTIBILIDAD	96
3.7 CONCLUSIONES.	103
CONCLUSIONES.	104
RECOMENDACIONES	105
REFERENCIAS BIBLIOGRÁFICAS.	106
BIBLIOGRAFÍA.	110
ANEXOS	114

ÍNDICE DE TABLAS

TABLA 1.1 PROGRAMAS ANALIZADOS.	17
TABLA 2.1 ACTORES DEL SISTEMA.	48
TABLA 2.2 CASO DE USO: AUTENTICAR.	52
TABLA 2.3 CASO DE USO: RECUPERAR CONTRASEÑA.	53
TABLA 2.4 CASO DE USO: CAMBIAR CONTRASEÑA.	53
TABLA 2.5 CASO DE USO: GESTIONAR INDICADOR.	54
TABLA 2.6 CASO DE USO: GESTIONAR TAREA.	55
TABLA 2.7 CASO DE USO: GESTIONAR VINCULACIÓN DE TRABAJADORES.	55
TABLA 2.8 CASO DE USO: GESTIONAR ASIGNACIÓN DE RECURSOS.	56
TABLA 2.9 CASO DE USO: VISUALIZAR FASE.	56
TABLA 2.10 CASO DE USO: GESTIONAR ETAPA.	57
TABLA 2.11 CASO DE USO: GESTIONAR FASE.	57
TABLA 2.12 CASO DE USO: GESTIONAR PERSONAL.	58
TABLA 2.13 CASO DE USO: GESTIONAR FLUJO.	59
TABLA 2.14 CASO DE USO: GESTIONAR PERMISO.	59
TABLA 2.15 CASO DE USO: GENERAR REPORTE EXPORTABLE.	60
TABLA 2.16 CASO DE USO: VISUALIZAR RESPONSABILIDADES.	60
TABLA 2.17 CASO DE USO: MODIFICAR INDICADOR.	61
TABLA 2.18 CASO DE USO: REALIZAR SALVA DE SEGURIDAD AL SISTEMA DE GESTIÓN.	62
TABLA 2.19 CASO DE USO: CONSULTAR AYUDA.	62
TABLA 2.20 CASO DE USO: DIAGNOSTICAR EQUIPO.	63
TABLA 2.21 CASO DE USO: DESCARGAR UTILIDAD DE DIAGNÓSTICO DE EQUIPO.	63
TABLA 2.22 CASO DE USO: SUBIR DIAGNÓSTICO REALIZADO A EQUIPO... 64	64
TABLA 2.23 CASO DE USO: ELIMINAR DIAGNÓSTICO INNECESARIO.	65
TABLA 2.24 CASO DE USO: VISUALIZAR DIAGNÓSTICO REALIZADO A EQUIPO.	65
TABLA 2.25 CASO DE USO: DIAGNOSTICAR RED.	66
TABLA 2.26 CASO DE USO: REALIZAR SALVA DE SEGURIDAD AL REGISTRO DE DIAGNÓSTICOS.	67
TABLA 3.1 FACTOR DE PESO DE LOS ACTORES.	97
TABLA 3.2 FACTOR DE PESO DE LOS CASOS DE USO.	97
TABLA 3.3 FACTOR DE COMPLEJIDAD TÉCNICA.	98
TABLA 3.4 FACTOR DE AMBIENTE.	99
TABLA 3.5 HORAS – PERSONA.	100
TABLA 3.6 DISTRIBUCIÓN DEL ESFUERZO.	101

ÍNDICE DE FIGURAS

FIGURA 1.1 SERVIDORES WEB MÁS UTILIZADOS.	29
FIGURA 2.1 MODELO DEL DOMINIO	40
FIGURA 2.2 ACTORES DEL SISTEMA.	47
FIGURA 2.3 PAQUETES	48
FIGURA 2.4 CASOS DE USO DEL PAQUETE DE GESTIÓN.	50
FIGURA 2.5 CASOS DE USO DEL PAQUETE DE DIAGNÓSTICO.	51
FIGURA 3.1 CASO DE USO: AUTENTICAR.	69
FIGURA 3.2 CASO DE USO: RECUPERAR CONTRASEÑA.	70
FIGURA 3.3 CASO DE USO: CAMBIAR CONTRASEÑA.	71
FIGURA 3.4 CASO DE USO: GESTIONAR INDICADOR.	72
FIGURA 3.5 CASO DE USO: GESTIONAR TAREA.	73
FIGURA 3.6 CASO DE USO: GESTIONAR VINCULACIÓN DE TRABAJADORES.	74
FIGURA 3.7 CASO DE USO: GESTIONAR ASIGNACIÓN DE RECURSOS.	75
FIGURA 3.8 CASO DE USO: VISUALIZAR FASE.	76
FIGURA 3.9 CASO DE USO: GESTIONAR ETAPA.	77
FIGURA 3.10 CASO DE USO: GESTIONAR FASE.	78
FIGURA 3.11 CASO DE USO: GESTIONAR PERSONAL.	79
FIGURA 3.12 CASO DE USO: GESTIONAR FLUJO.	80
FIGURA 3.13 CASO DE USO: GESTIONAR PERMISO.	81
FIGURA 3.14 CASO DE USO: GENERAR REPORTE EXPORTABLE.	82
FIGURA 3.15 CASO DE USO: VISUALIZAR RESPONSABILIDADES.	83
FIGURA 3.16 CASO DE USO: MODIFICAR INDICADOR.	84
FIGURA 3.17 CASO DE USO: REALIZAR SALVA DE SEGURIDAD AL SISTEMA DE GESTIÓN	85
FIGURA 3.18 CASO DE USO: CONSULTAR LA AYUDA.	85
FIGURA 3.19 CASOS DE USO: DIAGNOSTICAR EQUIPO Y DESCARGAR UTILIDAD DE DIAGNÓSTICO DE EQUIPO.	86
FIGURA 3.20 CASO DE USO: SUBIR DIAGNÓSTICO REALIZADO A EQUIPO.	87
FIGURA 3.21 CASO DE USO: ELIMINAR DIAGNÓSTICO INNECESARIO.	88
FIGURA 3.22 CASO DE USO: VISUALIZAR DIAGNÓSTICO REALIZADO A EQUIPO	89
FIGURA 3.23 CASO DE USO: DIAGNOSTICAR RED.	90
FIGURA 3.24 REALIZAR SALVA DE SEGURIDAD AL REGISTRO DE DIAGNÓSTICOS.	91
FIGURA 3.25 MODELO LÓGICO DE DATOS.	95
FIGURA 3.26 MODELO FÍSICO DE DATOS.	95
FIGURA 3.27 DIAGRAMA DE IMPLEMENTACIÓN.	96

Introducción

La ciencia es una fuente compleja, en ella se encuentran soluciones novedosas a los problemas actuales de la humanidad y el mundo en si. Una ética científica correcta contribuye al adecuado uso de los arsenales del conocimiento; sin embargo cuando se toma el timón de la investigación con las motivaciones incorrectas resulta en una carrera egoísta cuyo fin no será ya más el beneficio de todos sino el personal.

La Informática es hoy la ciencia que rige el flujo y utilización del conocimiento. El soporte lógico sobre el cual se desarrolla es el software informático. Por muchos ha sido visto como una plataforma para el enriquecimiento personal y han encaminado sus esfuerzos a construir una dependencia global hacia sí mismos. Esta es la esencia del software privativo. Pagar para tener el derecho a usar, pero nunca compartir ni innovar.

El software libre trae nuevos paradigmas. Su propio nombre refleja su esencia. Los modelos de software libre son abiertos y gratuitos. Se trata de sistemas operativos y programas de gran calidad capaces de satisfacer las necesidades de todo tipo de usuarios, creados tradicionalmente por una comunidad global de activistas entusiastas y, más recientemente, por importantes empresas de todo el mundo.

Pero si el software libre es tan robusto y versátil, ¿por qué no acaba de imponerse a los sistemas propietarios, que asfixian con sus licencias a las economías locales? La respuesta está dada por el hecho de que la infraestructura informática ha estado tradicionalmente basada en software privativo. Es por esto que se concibe la migración a software libre como un proceso cuyo objetivo es la sustitución de las infraestructuras basadas en Software Propietario por otras con funciones equivalentes basadas en software libre.

En el mundo entero se está gestando un vuelco hacia esta nueva dirección. No es tan solo una cuestión moral sino también económica e inteligente. El poder utilizar dichos recursos de forma libre es una razón de gran peso para dar las espaldas a lo que por un tiempo fue la única alternativa y abrir los ojos a lo que se impone como La Alternativa para Hoy. Tal y como dijera Dominique Ouredrago, inspector de Naciones Unidas en el 2006: **“es el vehículo más adecuado para su desarrollo, especialmente en materia de salud, educación y comercio internacional”** (“Migracion,” s.d.).

Específicamente la industria cubana del software hoy en día se ve seriamente afectada por el bloqueo norteamericano impuesto a la isla por parte del gobierno de los Estados Unidos de América (EUA), el cual prohíbe usar en territorio cubano cualquier producto que provenga de empresas norteamericanas (Ing. Yoandy Pérez Villazón & Ing. Ramón Paumier Samón, 2009).

Por este motivo se dan sólidos pasos encaminados a reemplazar toda la estructura informática hacia una plataforma de software libre. Esta es una política del gobierno cubano que se ve reflejada en las diversas organizaciones nacionales. Particularmente en el Ministerio de la Educación Superior en el cual se impone asumir un papel de vanguardia ya que el mismo tiene como razón de ser la capacitación y preparación de profesionales. (Ing. Yoandy Pérez Villazón & Ing. Ramón Paumier Samón, 2009).

Como Institución cabecera de la educación superior en la provincia de Cienfuegos, el centro de estudios universitarios “Carlos Rafael Rodríguez” también se ha proyectado en este proceso de migración.

Hasta el momento el plan de estudios ha estado concebido sobre una estructura privativa donde el sistema operativo predominante es Windows, en sus variantes “Windows 2000 Profesional” y “Windows XP”. Además de esto, diversas aplicaciones no libres están instaladas sobre esta plataforma.

Producto de la ya tan arraigada dependencia al uso de estos sistemas se hace necesario concebir un plan que permita de forma gradual y estructurada la inserción de las tecnologías de software libre. Los seres humanos por naturaleza se resisten al cambio. Una migración bien concebida y planificada evitará o al menos disminuirá considerablemente la oposición por parte de los usuarios.

Se hace entonces necesario definir etapas y tareas a emprender para lograr dicho objetivo. El correcto cumplimiento de estas tareas debe ser supervisado de forma continua por los responsables.

Un proceso de este tipo demanda la gestión de gran cantidad de información. La correcta estructuración del trabajo así como la supervisión del progreso son vitales.

Problema a Resolver.

El análisis de la situación existente conduce a plantear el siguiente **problema**: “**La necesidad de gestionar la información generada en el proceso de migración a Software Libre en la Universidad de Cienfuegos Carlos Rafael Rodríguez**”.

Objeto de Estudio y Campo de Acción

El **objeto de estudio** de la presente investigación está relacionado con el Proceso de Migración a Software Libre en la Universidad de Cienfuegos.

El **campo de acción** se delimita en la creación de una herramienta informática para la gestión de la información generada en el Proceso de Migración a Software Libre.

Objetivos Generales y Específicos

Ante esta problemática se traza como **objetivo general**:

“Desarrollar una herramienta informática que facilite la gestión de la información generada durante el proceso de Migración al Software Libre en la Universidad de Cienfuegos. “

De forma específica se plantean como objetivos:

- ❖ Diseñar una Base de Datos que permita almacenar de manera consistente la información del proceso.
- ❖ Implementar funcionalidades para la evaluación y el control del proceso de migración.
- ❖ Implementar funcionalidades para el diagnóstico de los programas en uso, a través de la red.
- ❖ Diseñar una interfaz gráfica amigable y práctica para el sistema.
- ❖ Elaborar reportes y gráficos que muestren el estado real del proceso de migración.
- ❖ Validar los datos procesados para impedir la introducción de errores.

Las **tareas** a realizar para cumplir con los objetivos propuestos son:

- ✓ Estudio sobre el comportamiento del tema de migración al software libre en la actualidad.
- ✓ Análisis de aplicaciones relacionadas con el dominio del problema.
- ✓ Estudio sobre la forma en que se proyecta el proceso en la Universidad.
- ✓ Valoración sobre qué aspectos del proceso de migración serán informatizados.
- ✓ Valoración de los informes y reportes que serían útiles para el control del proceso de migración.
- ✓ Valoración sobre la forma en que se mostrarán estos informes y qué niveles de acceso definir para los mismos.
- ✓ Investigación sobre los temas de seguridad y restricción de acceso a la información.
- ✓ Obtención de los Requerimientos Funcionales.

- ✓ Obtención de los Casos de Uso del sistema, agrupándolos por paquetes.
- ✓ Documentación detallada del sistema.
- ✓ Elaboración de la ayuda del sistema.

Idea a defender.

"El desarrollo de una herramienta informática que facilite la gestión de la información relacionada con el proceso de migración al software libre contribuirá de manera notable a la ejecución del mismo. "

Aporte Práctico

El **aporte práctico** de esta investigación es el propio sistema en sí. Se dispondrá de una herramienta que permita gestionar la información del proceso de Migración a Software Libre. De esta manera se garantizará que la estrategia sea desarrollada de forma organizada y con un mayor control de la información. Dicha información será almacenada de forma persistente y segura a la vez que estará disponible para ser consultada y analizada por el personal involucrado en la migración para la toma de decisiones y la evaluación del proceso.

Estructuración del contenido del trabajo de diploma.

El documento fue desarrollado en tres capítulos con el propósito de una adecuada comprensión del mismo. A continuación se explica brevemente el contenido que engloba cada uno de ellos:

- Capítulo I. Fundamentación Teórica: En este capítulo se abordan fundamentalmente los temas relacionados con los conceptos asociados al dominio del problema, la descripción del objeto de estudio y de la solución propuesta, los sistemas automatizados existentes vinculados al campo de acción, las tendencias y/o tecnologías actuales así como la fundamentación de la metodología utilizada.

- **Capitulo II. Diseño de la Solución Propuesta:** En este capítulo se describe la solución propuesta utilizando la metodología RUP. Se realiza la descripción de las entidades y los conceptos principales, las relaciones entre los mismos, así como las reglas que rigen el dominio asociado al problema. Se presentan, además, los requerimientos funcionales y no funcionales, los actores y casos de uso del sistema con la respectiva descripción de cada uno.
- **Capitulo III. Implementación de la Solución Propuesta:** Se detalla la estructura interna del sistema a través de los diagramas de clases Web. Se presentan diversos aspectos referidos al diseño, el tratamiento de errores, los modelos lógico y físico de datos que utilizará la aplicación, así como un estudio sobre la factibilidad de la construcción de la solución propuesta.

Capítulo 1: Fundamentación Teórica.

1.1 Introducción.

En este capítulo se abordan temas relacionados con los conceptos asociados al dominio del problema, la descripción del objeto de estudio y del objeto de automatización, los sistemas automatizados existentes vinculados al campo de acción, las tendencias y/o tecnologías actuales, los sistemas Gestores de Bases de Datos y la fundamentación de la metodología utilizada.

1.2 Descripción del dominio del problema.

1.2.1 Software Libre y Migración

Hace algunos años una nueva alternativa a tomado fuerza en el mundo de las aplicaciones informáticas. Esta basada en la ética correcta en la cual todos son beneficiados. Su propio nombre refleja su esencia. Se trata del software libre.

El software libre trae nuevos paradigmas. Por ejemplo, se desarrolla como producto de un esfuerzo colectivo, no necesariamente coordinado, de miles de programadores esparcidos por el mundo, que trabajan de una manera voluntaria, descentralizada, no estructurada, no remunerada y paralela. Otro paradigma es que esta modalidad de desarrollo esta basada en la confianza, cumpliendo con los elementos básicos para el desarrollo del software: calidad, velocidad y sujeto a pruebas. Por lo tanto, su desarrollo no sigue necesariamente procedimientos o normas formales (como lo establecen las metodologías clásicas de desarrollo de software), sino que esta basado en practicas y costumbres aprendidas a través de la experiencia (Jose Aguilar, s.d.).

Se define como software libre aquel que una vez obtenido, puede ser usado, copiado, estudiado, modificado y redistribuido libremente. De modo más preciso, el software libre se refiere a cuatro libertades que poseen los usuarios del software:

0. La libertad de usar el programa, con cualquier propósito.
1. La libertad de estudiar cómo funciona el programa, y adaptarlo a las necesidades personales. El acceso al código fuente es una condición previa para esto.
2. La libertad de distribuir copias, con lo que se puede ayudar al vecino.
3. La libertad de mejorar el programa y hacer públicas las mejoras a los demás, de modo que toda la comunidad se beneficie (Marcos Pérez Ramírez, 2003).

A la posibilidad de acceder abiertamente al código fuente del programa se le ha denominado **Open Source**, en español, Código Abierto.

La idea que late detrás del Open Source es bien sencilla: cuando los programadores en Internet pueden leer, modificar y redistribuir el código fuente de un programa, éste evoluciona, se desarrolla y mejora. Los usuarios lo adaptan a sus necesidades, corrigen sus errores a una velocidad impresionante, mayor a la aplicada en el desarrollo de software convencional o cerrado, dando como resultado la producción de un mejor software (Iñaki I. Rojo , 2009).

Pero si el software libre es tan robusto y versátil, ¿por qué no acaba de imponerse a los sistemas propietarios, que asfixian con sus licencias a las economías locales y no permiten su mejora por parte de terceros al negar el acceso a sus códigos fuente?

La respuesta está dada por el hecho de que la infraestructura informática ha estado tradicionalmente basada en software privativo. A esto se suman las presiones de las Entidades pro-Microsoft, y de una manera particularmente contradictoria se tiene la costumbre ya arraigada de usar Windows como sistema operativo. Es por esto que se concibe la migración a software libre como un proceso cuyo objetivo es la sustitución de las infraestructuras basadas en software propietario por otras con funciones equivalentes basadas en software libre. Esta

migración proporciona beneficios tanto económicos como prácticos (“Migracion,” s.d.).

La magia del software libre, en gran parte, consiste en el modelo de desarrollo abierto. Este modelo ha dado resultados muy interesantes, entre los cuales se cuentan proyectos y aplicaciones que, siendo software libre y de código abierto, admiten la ejecución en entornos no libres, como Microsoft Windows.

Todo esto en suma contribuye a la concepción de la Migración como un proceso gradual con características de implementación bien concebidas.

1.2.2 Panorama Mundial.

En el mundo entero se está gestando un vuelco hacia esta nueva dirección. No es tan solo una cuestión moral sino también económica e inteligente.

Algo está cambiando. Los usuarios domésticos, las empresas y las administraciones públicas se están dando cuenta de lo importante que es utilizar sistemas abiertos para la gestión y transmisión de la información. Hasta la ONU parece estar por la labor: según Dominique Ouredrago, inspector de Naciones Unidas en el 2006, en la segunda Conferencia Internacional de Software Libre celebrada en Málaga, “la ONU recomienda la utilización y el fomento del software libre tanto en el seno de la organización como entre los países miembros” porque **“es el vehículo más adecuado para su desarrollo, especialmente en materia de salud, educación y comercio internacional”** (Ing. Yoandy Pérez Villazón & Ing. Ramón Paumier Samón, 2009).

Cada vez son más los que hoy en día se suman al movimiento de software libre. En un artículo publicado en el sitio regiondigital.com el 13 de Mayo del 2008 se plantea que más de 50 millones de estudiantes brasileños utilizarán tecnologías abiertas a partir del 2009 (“El Gobierno de Brasil ha dado un paso más en favor de las tecnologías de código abierto,” 2008). Una publicación más reciente anuncia como usando una tecnología de "virtualización" sobre GNU/Linux, una computadora

podrá brindar servicio a 10 personas. De esta manera se crearán 356 mil puestos en lo que denominan "la mayor experiencia de virtualización del mundo" ("Software Libre en Venezuela," 2009).

Es de destacar que en Alemania no solo la administración y las universidades usan software libre, las empresas alemanas hace ya tiempo que vienen apostando por el código libre. Según una reciente encuesta el país europeo estaría a la cabeza en el uso de este tipo de software en todo el mundo ("Alemania es el país con mayor uso de software libre ," 2009).

En Venezuela se plantea que el modelo de software libre informático debe ser usado por todas las dependencias estatales, según la reciente publicación en Gaceta Oficial de las Normas Técnicas que regirán las Tecnologías de Información (TI) ("Software Libre en Venezuela," 2009).

A la lista se suman diversos países como son Ecuador, Israel y otros tantos que han visto el potencial que emerge de este movimiento (Yudivián Almeida Cruz, 2008)(Andrew Orłowski in San Francisco, 2003).

1.2.4 Software Libre en la Educación

Richard Stallman, considerado como el "padre del Software Libre", en una conferencia impartida en la Universidad de Palma de Mallorca en el 2004 planteó que el software libre en el aula permite aprender:

- Que no todo está hecho.
- Que aún hay retos y que las cosas siempre se pueden mejorar.
- A adoptar una postura constructiva.
- A cooperar con la comunidad local e internacional, sin distinción de edades, razas, nivel social, títulos, etc.
- Que lo mejor para cada quién/comunidad se escoge libremente de acuerdo a las necesidades/expectativas de cada sujeto o grupo social, y no debe ser impuesto por casas comerciales o estándares externos.

- La posibilidad de aprender de otros y que otros pueden aprender de nosotros.
- A propagar el conocimiento de forma libre (básicamente, lo que se trata de hacer cuando uno da una clase).
- La no discriminación.
- A trabajar en equipo.
- La libertad de investigar, crear, modificar y aprender.

En definitiva, el software libre en nuestras escuelas y en nuestra sociedad es otro modelo cultural y tecnológico, otra manera de entender la relación entre las personas, el conocimiento y las herramientas digitales.

Pero si se hace con los mismos criterios y de la misma forma que se está utilizando ahora respecto al software cerrado y privativo, se habrá perdido una magnífica ocasión de transformar nuestras aulas y nuestra sociedad.

Por otra parte surge la inquietud: ¿Se podrían hacer las mismas cosas que con el software privativo?

Actualmente no sólo se pueden hacer las mismas cosas con ambos tipos de software, libre y privativo, sino que en determinadas circunstancias se pueden hacer más cosas con el software libre. Además, al software libre, por ahora, no le afectan los virus y tiene un mayor nivel de seguridad en su uso con un menor consumo de recursos del computador por lo que no se necesita el último ordenador del mercado ni el más costoso para utilizarlo (José Luis Murillo, 2009).

1.2.4 Software Libre en Cuba

Una de las razones principales por las que nuestro país abraza el software libre es sin dudas el embargo de Estados Unidos que impide a Cuba comprar o actualizar productos como Windows, el sistema operativo más popular del planeta instalado en la isla a partir de copias piratas.

Además, el software privativo es un arma de doble filo, pues los fabricantes pasan sus códigos a las agencias de seguridad. La existencia de elementos espías como son las puertas traseras o “backdoor” no son un secreto para nadie.

La presentación del sistema operativo Nova en la Feria Informática 2009 desarrollada recientemente en La Habana, es un agigantado paso en la Migración Cubana. Nova resulta una distribución cubana de Sistema Operativo basada en GNU/Linux. Posee varias versiones, entre ellas, la orientada a las llamadas "terminales sin disco", y aquella destinada a la docencia, creada para los laboratorios de la UCI (Lianet Arias Sosa, 2009).

Desde hace ya un tiempo se dan sólidos pasos encaminados a reemplazar toda la estructura informática hacia una plataforma de software libre. Esta es una política del gobierno cubano que se ve reflejada en las diversas organizaciones nacionales.

1.2.4 Software Libre en las Universidades Cubanas

Las universidades son la fuente de investigación y conocimientos más grandes del país, hoy se cuenta con 3150 sedes universitarias y 65 escuelas de altos estudios por todo el país. La incorporación del software libre en estos centros sin duda alguna sería el paso más grande que daría el país en materia de migración. Si bien es cierto que nuestro país está resuelto a incorporarse de lleno y desplazar de manera creciente todo lazo o dependencia al software privado, también urge la necesidad de contar con los mejores métodos y herramientas que permitan facilitar la implantación del software libre (Ing. Yoandy Pérez Villazón & Ing. Ramón Paumier Samón, 2009).

El MES, (Ministerio de la Educación Superior), se impone asumir un papel de vanguardia ya que el mismo tiene como razón de ser la capacitación y preparación de profesionales. Además, está el hecho de que las universidades son los centros de investigación científica que repercuten de manera más contundente en el

desarrollo científico de una nación. Con este fin se ha presentado la METODOLOGÍA DE MIGRACIÓN A SOFTWARE LIBRE PROPUESTA, a partir de experiencias tanto positivas como negativas, nacionales e internacionales (Ing. Yoandy Pérez Villazón & Ing. Ramón Paumier Samón, 2009).

1.3 Objeto de estudio

1.3.1 Descripción General

La Universidad de Cienfuegos “Carlos Rafael Rodríguez” se encuentra ubicada en el municipio de Cienfuegos de dicha provincia, sito en carretera a Rodas, Km. 4, Cuatro Caminos; subordinado al Ministerio de Educación Superior. Dicha Institución tiene como visión **la formación de profesionales integrales comprometidos con la ideología de la Revolución Cubana. Participa protagónicamente en la transformación y desarrollo de la Provincia y del País a través de la introducción y generalización de los resultados de la Ciencia y la Técnica, de la Extensión Universitaria y de la Superación y desarrollo de los profesionales y dirigentes.** En cuanto a su objeto social podemos señalar que la Universidad se propone: **Llevar a cabo la formación continua de profesionales, en las ramas de las Ciencias Técnicas, Agropecuarias, Económicas, Sociales y Humanistas, Exactas y Naturales y de la Cultura Física, así como de la actividad científica técnica y la extensión universitaria.** Entre otros aspectos la visión de este importante centro de estudios incluye ser **líder en la contribución al desarrollo económico social de la provincia, estar en la avanzada de las Centro Educación Superior del país.**

La Incorporación del software libre complementa de forma notable estas características propias de la entidad en cuestión. El software libre trasciende de ser una mera tecnología y se presenta además como una perfecta oportunidad para educar y formar valores necesarios en la sociedad actual. La idea fundamental que radica en este movimiento es la solidaridad, la libertad y el

compartir sin otro interés más que el beneficio colectivo y el desarrollo de todos los participantes.

A diferencia del software privativo, con el software de código abierto se tiene una enorme oportunidad para la investigación y la innovación, lo que garantiza una formación de profesionales más capaces. Ahora cada aplicación puede ser estudiada a fondo, adecuada y perfeccionada de acuerdo a las necesidades propias.

El software libre es un asunto de actualidad y promete imponerse gradualmente en la industria informática como una herramienta de evolución continua lo que ya le amerita logros en aplicaciones que dejan rezagadas a sus homólogas privativas. No puede un centro cuya naturaleza implica la introducción y generalización de los resultados de la Ciencia y la Técnica en la sociedad estar enajenado a este fenómeno.

Hasta el momento el plan de estudios se ha concebido sobre una estructura privativa donde el sistema operativo predominante es Windows en sus variantes "Windows 2000 Profesional" y "Windows XP".

Dadas las probadas ventajas del software libre, el MES (Ministerio de la Educación Superior) se proyecta en aras de difundir su utilización en sus dependencias. Es por esto que la Universidad de Cienfuegos trabaja en una estrategia y ya se han dado pasos concretos con este fin.

Dada la establecida familiarización con los sistemas privados y la dependencia que ha generado el exclusivo uso de los mismos se hace necesario concebir un plan que mediante pasos definidos inserte gradualmente las tecnologías de software libre hasta un punto donde sea desplazada de forma total la dependencia a algún programa privativo.

En este propósito se van ejecutando acciones bien concebidas tales como:

- Se ha modificado el plan de estudios de los estudiantes que ingresan actualmente a la carrera de informática.
- Se ha procedido a la instalación de GNU/Linux en algunos de los terminales disponibles en los laboratorios docentes.
- También varios servicios corren actualmente sobre GNU/Linux, como son la intranet, red y correo electrónico.
- Varios Trabajo de Diploma han tomado la temática del software libre como propósito de investigación y desarrollo, tributando así al Centro de Estudios como también al país.

1.4 Sistemas existentes vinculados al campo de acción.

Se realizó un estudio de los Sistemas automatizados existentes con el fin de hallar una herramienta que permitiera la gestión de la información del Proceso de Migración así como el diagnóstico a redes y equipos.

En la parte concebida para diagnosticar y controlar los sistemas instalados en la red, es importante señalar tres programas concebidos con este fin los cuáles fueron estudiados pero no incorporados. La tabla a continuación presenta una descripción de los mismos y la razón por la cuál no fueron utilizados.

Nombre	Descripción	Razones para no usarlo
NMAP	Nmap ('Network Mapper'), es una herramienta open source, diseñada para explorar y para realizar auditorias de seguridad en una red de computadoras. Esta herramienta pone al descubierto los puertos abiertos en las computadoras de una red. También es posible conocer como se encuentra organizada, y de cuantas computadoras consta una red ("Introducción a Nmap,"	- Cuando se trata de múltiples maquinas los resultados no se corresponden con la realidad y en la mayoría de los casos no ofrece información sobre los sistemas operativos después de analizar el primer equipo en un

	2003).	rango.
XPROBE	Herramienta de identificación de sistemas operativos. XProbe es una herramienta que sirve para determinar el sistema operativo de un equipo remoto. Logra esto utilizando algunas de las mismas técnicas que Nmap al igual que muchas ideas diferentes ("Fundación Código Libre Dominicano," s.d.). Muestra el por ciento de certeza en su diagnóstico (Ofir Arkin & Fyodor Yarochkin, 2002).	- Debe ser instalado sobre GNU/Linux. Uno de los requerimientos para la solución al problema en cuestión es que pueda ser montado tanto en este Sistema Operativo como en Windows.
OCS INVENTORY	Es una potente aplicación diseñada para ayudar al administrador del sistema o red a mantener la pista de la configuración y el software instalado en los computadores de la red. Su funcionamiento está basado en agentes que se instalan en los terminales clientes y tributan información al servidor.	Es un programa muy bien concebido para la administración de redes. Su implementación requiere la instalación de agentes en las máquinas.

Tabla 1.1 Programas analizados.

Aunque estos programas no se adecuaran al propósito del trabajo en cuestión, se recomienda su uso para razones específicas y por personal con conocimientos en redes de computadoras. Los mismos son aplicaciones libres, particularmente útiles en la detección de intrusos y fallas de seguridad en la red.

Estos sistemas no son nunca cien por ciento confiables, aún NMAP siendo una de las herramientas preferidas para la detección de sistemas operativos puede ser engañado (David Barroso Berrueta, 2003). Los programas descritos anteriormente están basados en el "fingerprinting" (Huella o huella digital) de la capa de red del

host remoto. Conceptualmente, el engaño es sencillo, basta con modificar la capa de red del Kernel, y especificar como se debe responder ante determinados paquetes raros. Así, cambiar las huellas del sistema operativo y parecer otro (Gerard Farras, 2009).

Dadas estas circunstancias se optó por una solución más sencilla y efectiva en referencia a la problemática.

La utilidad ping comprueba el estado de la conexión con uno o varios equipos remotos por medio de los paquetes de solicitud de eco y de respuesta de eco. Es útil para diagnosticar los errores en redes o enrutadores IP. El valor del TTL (time to live), tiempo a vivir, de un paquete de ip representa el número máximo de enrutadores a través de los cuales el mismo puede viajar. En la práctica actual se espera de cada enrutador en Internet un decremento al valor del TTL exactamente en una unidad por lo que hoy se recomiendan valores por encima de 64 (“Default TTL Values in TCP/IP,” 1999).

Distintos sistemas operativos inicializan el campo TTL con distintos valores (ANEXO 1).

El campo TTL se puede utilizar para (Fernando Gont, 2008):

- Identificar el sistema operativo utilizado por un sistema
- Identificar sistemas físicos detrás de una dirección IP
- Hallar un sistema en la topología de red

Al igual que en el caso de los programas descritos anteriormente, no se trata de un método infalible, pero a demostrado ser, de las opciones estudiadas, la más apropiada para el diagnóstico de sistemas operativos en la solución propuesta.

Pero es en el aspecto organizativo y de gestión de la información donde se hace sentir con más fuerzas la necesidad de un sistema automatizado. La estructura concebida particularmente en el Ministerio de Educación Superior requiere que

dicha herramienta se ajuste a su proyección. Debe además mostrar flexibilidad para ser adaptada a otras estructuras, como en el caso de la Universidad de Cienfuegos.

No existe una aplicación con estas características y menos una que incorpore la gestión de la información y el diagnóstico de programas en una misma herramienta.

1.5 Descripción del Objeto de Informatización.

El Sistema de Apoyo a la Migración (SAM) se presenta como una herramienta adecuada para la Gestión y Control del Proceso de Migración a Software Libre en la Universidad de Cienfuegos “Carlos Rafael Rodríguez”.

El mismo está ajustado a las particularidades de nuestra Universidad, aunque presenta características que lo hacen flexible para su implantación en otros centros.

Permite una correcta gestión de las fases, etapas y tareas que se planifiquen así como de los flujos de trabajo en que estas se desarrollan. Los responsables de fases y etapas podrán planificar y controlar las tareas que se deben acometer de forma ágil y organizada.

El sistema es capaz de realizar diagnósticos para determinar el tipo de sistema operativo instalado en los ordenadores de la red, así como la naturaleza de los demás programas de aplicación existentes en los equipos con sistema operativo Windows.

En general tenemos como consideración fundamental que este sistema incorpora en uno ambas cuestiones: gestión de la información del proceso de migración y diagnóstico de programas en redes y equipos.

1.6 Tendencias, metodologías y/o tecnologías actuales

1.6.1 Introducción

En el desarrollo de una herramienta de software, las etapas deben estar soportadas por las indicaciones de alguna metodología para poder garantizar su calidad. Se debe tener presente, además, que antes de elaborarse cualquier aplicación es necesario realizar un estudio de las tecnologías actuales con el objetivo de seleccionar la más adecuada según los requerimientos que se deben cumplir para el desarrollo de la nueva propuesta.

1.6.2 Fundamentación de la Metodología Utilizada

Rational Unified Process (RUP)

Es un proceso de desarrollo de software que comprende un conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema informático. Es un marco de trabajo genérico el cual puede especializarse, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyectos.

Fue creado por un grupo de estudiosos de la Ingeniería de Software formado por: Ivar Jacobson, Grady Booch y James Rumbaugh en el año 1998. Es un proceso basado en componentes y utiliza el Lenguaje Unificado de Modelado (UML) para preparar todos los esquemas de un sistema software. No obstante, los verdaderos aspectos definitorios de RUP se resumen en tres frases clave: está dirigido por casos de uso, centrado en la arquitectura y es iterativo e incremental.

Además cubre el ciclo de vida de un proyecto y toma en cuenta las mejores prácticas a utilizar en el modelo de desarrollo de software.

A continuación se muestran estas prácticas:

- Desarrollo de software en forma iterativa.

- Manejo de requerimientos.
- Utiliza arquitectura basada en componentes.
- Modela el software visualmente.
- Verifica la calidad del software.
- Controla los cambios.

Esta herramienta integra todos los elementos que propone la metodología para cubrir el ciclo de vida de un proyecto (Ivar Jacobson, Grady Booch, & Jame Rumbaugh, s.d.).

Se decidió utilizar esta metodología porque:

Es una plataforma flexible que usa reglas de negocio y procedimientos comerciales probados los cuales son conocidos como “mejores prácticas”. Es una guía de cómo utilizar de manera efectiva UML. Genera y mantiene modelos en vez de gran cantidad de papeles de documentación, como es el caso de MSF (Microsoft Solution Framework), otra metodología tradicional pero que además limita al cliente en las herramientas de desarrollo.

El RUP unifica el equipo de trabajo con procesos comunes optimizando la comunicación y el entendimiento para todas las tareas, responsabilidades y artefactos.

Unified Modeling Language (UML)

UML (Unified Modeling Language) es un lenguaje que permite modelar, construir y documentar los elementos que forman un sistema software orientado a objetos. Se ha convertido en el estándar de facto de la industria, debido a que ha sido impulsado por los autores de los tres métodos más usados de orientación a objetos: Grady Booch, Ivar Jacobson y Jim Rumbaugh. Estos autores fueron contratados por la empresa Rational Software Co. para crear una notación unificada en la que basar la construcción de sus herramientas CASE. En el

proceso de creación de UML han participado, no obstante, otras empresas de gran peso en la industria como Microsoft, Hewlett-Packard, Oracle o IBM, así como grupos de analistas y desarrolladores (Xavier Ferré Grau & María Isabel Sánchez Segura, 2009).

Se decide utilizar UML porque es conocido como el lenguaje estándar en el análisis y diseño de sistemas de cómputo. Mediante UML es posible establecer los requerimientos y las estructuras necesarias para representar un sistema de software antes de escribir código.

1.6.3 Tendencias Actuales y Tecnologías a Considerar

1.6.3.1 Arquitectura de Desarrollo de N Capas

Las aplicaciones en 'n' niveles surgieron por primera vez como una forma de resolver algunos de los problemas asociados a las aplicaciones cliente/servidor tradicionales (modelo de dos capas), pero con la llegada de la Web, esta arquitectura ha llegado a dominar el nuevo desarrollo.

La premisa base para la separación de capas es que: Distintas Responsabilidades no deben ser delegadas en la misma clase. (Separación de incumbencias). La tendencia actual en aplicaciones WEB es la Arquitectura de n-capas. El modelo más básico es el de tres capas:

Capa de Presentación

Capa de Negocio

Capa de Datos

Capa de Presentación: Comprende las responsabilidades de lógica de presentación:

- Navegabilidad del sistema

- Validación de datos de entrada
- Formateo de los datos de salida
- Internacionalización
- Renderizado de presentación

Capa de Negocio: Comprende las responsabilidades de lógica de negocio (o dominio) del sistema. Esta capa reúne todos los aspectos del software que automatizan o apoyan los procesos de negocio que llevan a cabo los usuarios. Estos aspectos típicamente incluyen las tareas que forman parte de los procesos, las reglas y restricciones que aplican.

- Resultado del análisis funcional: Conjunto de reglas de negocio que abstraen el mundo real.
- La capa de negocio ha de ser independiente de la capa de presentación y viceversa (en la medida de lo posible).

Capa de Acceso a Datos: Comprende las responsabilidades de lógica de persistencia de las entidades que maneja el sistema en desarrollo. Se encarga de almacenar, recuperar y mantener los datos así como de su integridad.

- Inserción
- Eliminación
- Actualizaciones
- Búsquedas
- Etc.

No tiene porqué tratarse necesariamente de una base de datos relacional. Estos datos consisten en cualquier fuente de información, incluido una Base de datos, un conjunto de documentos XML o incluso un servicio de directorio (“Arquitectura Web,” 2009).

El modelo de N capas persigue, con su arquitectura, que las aplicaciones maximicen aspectos trascendentes en el desempeño como son:

- **Autonomía:** Habilidad de una aplicación para gobernar sus recursos críticos.
- **Confiabilidad:** Habilidad de una aplicación para proporcionar resultados exactos.
- **Disponibilidad:** Cantidad de tiempo que una aplicación es capaz de dar servicio confiablemente a las peticiones del cliente.
- **Escalabilidad:** Meta utópica del crecimiento lineal del rendimiento al agregar recursos adicionales, y es lo que le permite a una aplicación servir desde 10 usuarios, hasta decenas de miles de usuarios, simplemente agregando o quitando recursos como sea necesario para "escalar" la aplicación.
- **Interoperabilidad:** Habilidad de una aplicación para acceder a las aplicaciones, los datos o los recursos en otras plataformas.

El uso de una arquitectura de N capas permite que la potencia de cálculo recaiga en el servidor. De esta manera, los clientes son cada vez más ligeros y no necesitan ni demasiadas capacidades de cálculo ni un excesivo software instalado, porque la capa de negocio y la de datos se encuentran centralizadas en el servidor (Oscar Muñoz, 2007).

1.6.3.2 Tecnologías Web

Las tecnologías Web juegan un importante papel en el desarrollo informático actual. La Internet se ha ido desarrollando progresivamente para convertirse en un ambiente donde se implementan potentes aplicaciones cliente/servidor o arquitecturas de n capas, unido a ello han ido surgiendo nuevas tecnologías que se relacionan con el desarrollo Web lo que hacen a éste más interactivo e interesante. Entre las tecnologías utilizadas para la creación y mantenimientos de sitios Web, están las que funcionan del lado del cliente y las del lado del servidor.

LADO DEL CLIENTE

LENGUAJE HTML (HyperText Markup Language)

Lenguaje basado en marcas o etiquetas diseñado para la creación y publicación de páginas Web. HTML es una sintaxis para definir los elementos que se quieren mostrar en una página, y su estructura y ubicación, así como las relaciones entre ellos y con otros elementos de la Web mediante hipervínculos. El lenguaje HTML es un estándar del W3C (Iván Nieto Pérez, 2009).

CSS (Cascading Style Sheets)

Las hojas de estilo en cascada o CSS constituyen el estándar para la inserción de estilos (tamaños, colores, tipografías, espacios, bordes, etc.) a documentos estructurados, como por ejemplo, páginas HTML o XML. El objetivo de la definición de este estándar del W3C es permitir la separación entre las normas de presentación y el propio contenido a mostrar (Iván Nieto Pérez, 2009).

LENGUAJE JAVASCRIPT

JavaScript es un lenguaje de programación interpretado, es decir, que no requiere compilación, utilizado principalmente en páginas Web, con una sintaxis semejante a la del lenguaje Java y el lenguaje C. Al igual que Java, JavaScript es un lenguaje orientado a objetos propiamente dicho. Todos los navegadores modernos interpretan el código JavaScript integrado dentro de las páginas Web. Para interactuar con una página Web se provee al lenguaje JavaScript de una implementación del DOM (Modelo de Objetos del Documento).

El lenguaje fue inventado por Brendan Eich en la empresa Netscape Communications, que es la que desarrolló los primeros navegadores Web

comerciales. Apareció por primera vez en el producto de Netscape llamado Netscape Navigator 2.0.

Los autores inicialmente lo llamaron Mocha y más tarde LiveScript pero fue rebautizado como JavaScript en un anuncio conjunto entre Sun Microsystems y Netscape, el 4 de diciembre de 1995 (“JavaScript - Wikipedia, la enciclopedia libre,” 2009).

Se elige JavaScript para efectuar las validaciones y otras interacciones requeridas en el lado del cliente ya que este lenguaje es interpretado por todos los navegadores modernos. Así se contribuye a un sistema multiplataforma.

AJAX (Asynchronous JavaScript)

AJAX es el acrónimo inglés para Asynchronous JavaScript and XML (JavaScript y XML asíncrono). AJAX no es una tecnología; AJAX es una técnica de desarrollo web que genera aplicaciones Web interactivas combinando:

- Document Object Model (DOM) para visualizar dinámicamente e interactuar con la información presentada.
- XML, XSLT para intercambiar y manipular datos.
- CSS para definir el aspecto (look and feel) del documento.
- JSON y JSON-RPC pueden ser alternativas a XML/XSLT
- XMLHttpRequest para recuperar datos de forma asíncrona.
- Javascript como nexo de unión de todas estas tecnologías.

AJAX es un patrón de diseño que propone un nuevo modelo de interacción Web combinando las tecnologías anteriores.

Se optó por esta técnica en algunos casos pues en aplicaciones AJAX se envían peticiones vía http(s) mediante eventos, scripts o rutinas al servidor Web, para

obtener únicamente la información necesaria. Esto redundará en una mayor interacción gracias a la reducción de información intercambiada entre servidor y cliente, y a que parte del proceso de la información se hace en el propio cliente, liberando al servidor de ese trabajo. Además la petición se realiza como proceso de fondo (background), por lo que el usuario no tiene que esperar que el proceso concluya en su totalidad para continuar interactuando con la aplicación (Yoandry Pacheco Aguila, 2007).

NAVEGADOR INTERNET EXPLORER

Anteriormente Microsoft Internet Explorer; abreviado MSIE, comúnmente abreviado como IE, es un navegador web producido por Microsoft para el sistema operativo Windows desde 1995 y más tarde para Sun Solaris y Apple Macintosh, estas dos últimas discontinuadas en el 2002 y 2006 respectivamente. Ha sido el navegador Web más utilizado desde 1999, con un pico sostenido de cuota de utilización durante el 2002 y 2003 del 95% en sus versiones 5 y 6. Esa cuota de mercado ha disminuido paulatinamente debido a una renovada competencia por parte de otros navegadores, principalmente Mozilla Firefox. Microsoft gastó más de 100 millones de dólares (USD) al año en el decenio de 1990, con más de 1000 personas trabajando en IE para 1999.

Su versión más reciente es la 8.0, la cual está disponible gratuitamente como actualización para Windows XP Service Pack 2, Windows Server 2003 con Service Pack 1 o posterior, Windows Vista, y Windows Server 2008. (“Internet Explorer - Wikipedia, la enciclopedia libre,” 2009).

NAVEGADOR MOZILLA FIREFOX

Es un navegador de Internet libre y de código abierto descendiente de Mozilla Application Suite, desarrollado por la Corporación Mozilla, la Fundación Mozilla y un gran número de voluntarios externos.

Firefox es un navegador multiplataforma y está disponible en varias versiones de Microsoft Windows, Mac OS X, GNU/Linux y algunos sistemas basados en Unix. Su código fuente es software libre, publicado bajo una triple licencia GPL/LGPL/MPL.

Cuenta con el 22.05% del mercado de navegadores Web en marzo del 2009, por lo que es el segundo navegador más popular en todo el mundo, después de Internet Explorer.

Incluye navegación por pestañas, corrector ortográfico, búsqueda progresiva, marcadores dinámicos, un administrador de descargas y un sistema de búsqueda integrado que utiliza el motor de búsqueda que desee el usuario. Además se pueden añadir funciones a través de complementos desarrolladas por terceros (“Mozilla Firefox - Wikipedia, la enciclopedia libre,” 2009).

LADO DEL SERVIDOR

SERVIDOR APACHE

Apache es el servidor Web más utilizado hoy en día, superando todas las versiones comerciales que existen. Esto queda demostrado según estudios realizados por Netcraft, que han determinado que cerca del 60% de los sitios activos están soportados por Apache.

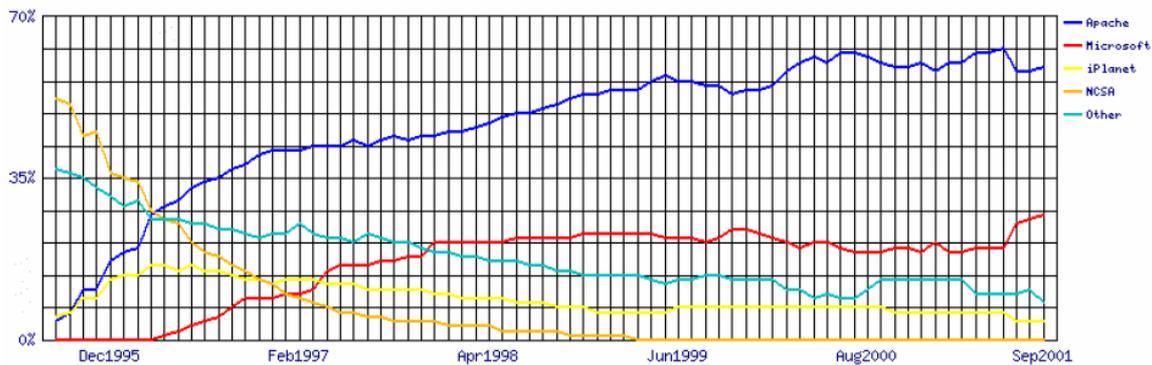


Figura 1.1 Servidores Web más utilizados.

Apache deriva de un servidor HTTP desarrollado por Rob McCool de la NCSA (National Center for Supercomputing Applications). Posteriormente, tomando como base el HTTP de NCSA se desarrollaron extensiones y solucionaron bugs detectados teniendo como objetivo construir un servidor más rápido, robusto y eficiente. Fue así que la primera versión oficial de Apache fue liberada en abril de 1995. Hoy, con un enorme equipo de voluntarios al rededor del mundo se han logrado estos objetivos logrando superar a grandes compañías como Microsoft y Netscape.

Entre sus características destacan:

- Multiplataforma, ha sido desarrollado para plataforma WinXX, Netware, Unix y Linux.
- Su desarrollo ha sido de acuerdo al protocolo HTTP/1.1 normalizado por el W3C (WWW Consortium)
- Modular, puede ser adaptado a diferentes entornos y necesidades, con los diferentes módulos de apoyo que proporciona, y con la API de programación de módulos, para el desarrollo de módulos específicos
- Open Source
- Extensible, gracias a ser modular se han desarrollado diversas extensiones entre las que destaca PHP (Aileen Morrison, 2009).

Se decidió utilizar Apache pues ha demostrado ser un servidor seguro y confiable, de código abierto y gratuito. Siendo el servidor Web más utilizado hoy en día.

SMTP (Simple Mail Transfer Protocol)

Protocolo de comunicaciones que utiliza un servidor de correo para enviar los mensajes hacia otro servidor de correo. Es igualmente el protocolo que se utiliza entre el cliente de correo del emisor y su servidor de correo. Para recibir los mensajes, el cliente de correo del receptor se comunica con su servidor de correo utilizando los protocolos POP o IMAP (Iván Nieto Pérez, 2009).

LENGUAJE PHP (Hypertext Preprocessor)

PHP es un lenguaje de programación en servidor de tipo script. Su nombre es el acrónimo de Hypertext Preprocessor. Tiene la ventaja de ser multiplataforma (Unix/Linux/Windows) y de utilizar una sintaxis muy sencilla, por lo que su aprendizaje y uso es muy rápido, además de ser gratuito. Como características importantes tiene el amplio soporte con las bases de datos más importantes (Oracle, MySQL) o a través de ODBC, incluye también soporte para XML y LDAP. Está siendo la solución adoptada por un gran número de programadores y empresas para llevar a cabo sus proyectos en Internet (Esteban Trigos García, 2009).

Se optó por PHP como lenguaje de programación pues está soportado en la mayoría de las plataformas de Sistemas Operativos, mientras que otros como el ASP, por ser software propietario, no son multiplataforma.

El PHP no tiene costo oculto, o sea que cuando se adquiere incluye un sinnúmero de bibliotecas que proporcionan el soporte para la mayoría de las aplicaciones Web, por ejemplo e-mail, generación de ficheros PDF y otros. En caso de que no se tengan las bibliotecas estas se pueden encontrar gratis en Internet.

PHP es rápido, seguro, gratuito y multiplataforma.

1.6.3.3 Sistemas Gestores de Bases de Datos

LENGUAJE DE CONSULTA ESTRUCTURADO SQL

El SQL es un lenguaje de acceso a bases de datos que explota la flexibilidad y potencia de los sistemas relacionales permitiendo gran variedad de operaciones sobre los mismos.

El lenguaje de consulta estructurado (Structured Query Language) es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones sobre las mismas. Una de sus características es el manejo del álgebra y el cálculo relacional permitiendo lanzar consultas con el fin de recuperar -de una forma sencilla- información de interés de una base de datos, así como también hacer cambios sobre la misma (“SQL - Wikipedia, la enciclopedia libre,” 2009).

MySQL

El software MySQL® proporciona un servidor de base de datos SQL (Structured Query Language) muy rápido, multi-usuario y robusto. El servidor MySQL está diseñado para entornos de producción críticos, con alta carga de trabajo así como para integrarse en software para ser distribuido.

El software MySQL tiene una doble licencia. Los usuarios pueden elegir entre usar el software MySQL como un producto Open Source bajo los términos de la licencia GNU General Public License (<http://www.fsf.org/licenses/>) o pueden adquirir una licencia comercial estándar (“MySQL 5.0 Reference Manual,” 2009).

Se optó por MySQL porque es rápido, confiable y fácil de usar. Con MySQL es posible la manipulación de bases de datos muy grandes. Es un sistema multiplataforma de base de datos relacionales, lo que da velocidad y flexibilidad, cuenta con un sistema de contraseñas muy seguro que permite la autenticación básica para el acceso al servidor.

El lenguaje PHP es altamente compatible con MySQL, por el amplio conjunto de comandos definidos para el tratamiento de este. Corre tanto en maquinas con GNU/Linux como Windows (A demostrado ser mejor en GNU/Linux).

1.6.4 Herramientas de Desarrollo

NOTEPAD ++

Notepad++ es un editor gratuito de código fuente, que soporta varios lenguajes de programación y se ejecuta en MS Windows. Este proyecto, basado en el componente de edición Scintilla (un poderoso editor), está escrito en C++ utilizando directamente la API de win32 y STL (lo que asegura una velocidad mayor de ejecución y un tamaño más reducido del programa final), se distribuye bajo los términos de la Licencia Pública General de GNU.

Se escoge esta herramienta de desarrollo porque, en primer lugar es de licencia pública. Además, es una aplicación bien ligera y su uso prácticamente no consume recursos de memoria. Es capaz de detectar el lenguaje de programación usado a partir del código. Posee la capacidad de determinar el comienzo y el fin de tags HTML y bloques de código resaltando con colores la sintaxis del lenguaje usado.

SQL MANAGER FOR MySQL

El EMS Gerente de SQL para MySQL es una herramienta de alto rendimiento para administración de Servidores de Bases de Datos MySQL. El Manager admite cualquiera de las versiones de MySQL. Incluso las más modernas. Apoya las funcionalidades de MySQL pues trabaja con vistas, procedimientos almacenados, etc. Este software ofrece muchas herramientas poderosas para los usuarios con experiencia satisfaciendo así todas sus necesidades. El Manager tiene una interfaz grafica amigable por lo que cualquier usuario con poca experiencia puede acceder a esta aplicación y navegar de forma orientada por la misma (“EMS SQL Manager for MySQL,” 2008).

PHPMyAdmin

Es una herramienta escrita en PHP con la intención de manejar la administración de MySQL a través de páginas Web, utilizando Internet. Actualmente puede crear y eliminar Bases de Datos, crear, eliminar y alterar tablas, borrar, editar y añadir campos, ejecutar cualquier sentencia SQL, administrar claves en campos, administrar privilegios, exportar datos en varios formatos y está disponible en 50 idiomas. Se encuentra disponible bajo la licencia GPL. Este proyecto se encuentra vigente desde el año 1998, siendo el mejor evaluado en la comunidad de descargas de SourceForge.net como la descarga del mes de diciembre del 2002. Como esta herramienta corre en máquinas con Servidores Web y Soporte de PHP y MySQL, la tecnología utilizada ha ido variando durante su desarrollo (“PHPMyAdmin,” 2008).

PHPDESIGNER

phpDesigner es un completo entorno de desarrollo y programación especialmente diseñado para programadores de PHP, aunque también permite trabajar con comodidad en otros lenguajes de programación como HTML, XHTML, CSS y SQL.

Ofrece toda una serie de asistentes y diálogos integrados que facilitan en todo momento la tarea, además de acceso directo a librerías de código o scripts de uso habitual, utilidades diversas y toda suerte de herramientas, todo ello en una interfaz de diseño sencillo y elegante.

phpDesigner cuenta con cliente de FTP y navegador de ficheros integrado, utilidades de corrección y auto-completado, búsqueda integrada en Google y soporte para proyectos, además de usar un práctico esquema de color para la sintaxis del código fuente que facilita enormemente la programación (“phpDesigner,” s.d.).

phpDesigner soporta los siguientes formatos:

PHP, HTML, XHTML, CSS, Java, Perl, JavaScript, VB, C# y SQL

Se opta por esta utilidad fundamentalmente por su probada capacidad en el auto-completado de códigos lo cual la hace especialmente útil en algunas circunstancias.

BORLAND DELPHI 7.0

Borland Delphi surgió en Junio de 1995. Es un entorno de desarrollo visual para Windows basado en el lenguaje Object Pascal. Object Pascal es un lenguaje de programación que surge a partir del desarrollo de Borland Pascal 7.0 y, es por tanto totalmente compatible con él, por lo que se considera un lenguaje que ocupa un lugar muy importante en la programación. Otra característica importante de Object Pascal es su capacidad para admitir diferentes tipos de datos, entre los cuales se encuentran los datos de tipo simple, los de tipo estructurado y los de tipo puntero.

Delphi en aplicaciones RAD (Rapid Application Development) constituye lo más elevado, al combinar un entorno de desarrollo visual altamente productivo, un compilador muy veloz, un potente lenguaje orientado a objeto (POO), un abundante repertorio de herramientas fundamentalmente las especializadas cliente/servidor consideradas una de las mejores del mercado.

Delphi incluye una biblioteca de clases bien diseñada denominada Biblioteca de Componentes Visuales-Visual Component Library-(VCL) y en su versión 7, una jerarquía multiplataforma paralela denominada CLX. Estas jerarquías de objetos incluyen componentes visuales y no visuales, tales como los pertenecientes a la categoría de acceso a datos, con los que puede establecerse conexiones de forma nativa o mediante capas intermedias (como ADO, BDE u ODBC) a la mayoría de las bases de datos relacionales existentes en el mercado (Yolanda de J. Ofarril Dinza, 2009).

Se eligió Delphi para programar la utilidad de diagnóstico de Software Instalado en Terminales con Windows. Delphi cuenta con la librería requerida para manipular el

Registro. Dicha librería es propia de este sistema operativo por lo que no fue posible su confección en PHP.

RATIONAL ROSE

La complejidad de los proyectos de software hoy en día, el constante cambio de requerimientos y la falta de una documentación durante el proceso de desarrollo provoca que los proyectos se retrasen en tiempo y se incrementen en costo. La solución a esta problemática es implantar una arquitectura de desarrollo que permita hacer seguimiento a los proyectos desde su etapa de requerimientos, hasta su implantación.

Rational ofrece un Proceso Unificado (RUP) para el desarrollo de los proyectos de software, desde la etapa de Ingeniería de Requerimientos hasta la etapa de pruebas. Para cada una de estas etapas existe una herramienta que ayuda en la administración de los proyectos, Rose es la herramienta de Rational para la etapa de análisis y diseño de sistemas.

Rose es una herramienta con plataforma independiente que ayuda a la comunicación entre los miembros del equipo, a monitorear el tiempo de desarrollo y a entender el entorno de los sistemas. Una de las grandes ventajas de Rose es que utiliza la notación estándar en la arquitectura de Software (UML), la cual permite a los arquitectos de software y desarrolladores visualizar el sistema completo utilizando un lenguaje común. Otra ventaja de Rose es que los diseñadores pueden modelar sus componentes e interfaces en forma individual y luego unirlos con otros componentes del proyecto. Además Rose soporta la construcción de componentes en lenguajes como C++, Visual Basic, Java, Ada, genera IDL's para aplicaciones CORBA (“PROYECTOS,” s.d.).

Se escoge Racional Rose, además, por ser es la herramienta de Análisis, Diseño, Modelado y Construcción de software Orientado a Objetos líder en el mercado.

1.6.6 Conclusiones

A modo de conclusión se puede señalar la necesidad de desarrollar una solución para el problema que origina la presente investigación. Se ha podido comprobar la importancia de la Gestión de la información que se genera en un proceso de migración a software libre así como el diagnóstico de la naturaleza de los programas instalados. La ausencia de una herramienta con estas características implica la novedad que resultaría el confeccionar una aplicación informática que desarrolle los requerimientos involucrados en este proceso.

Capítulo 2: Descripción de la Solución Propuesta

2.1 Introducción.

En el presente capítulo se describe la solución propuesta utilizando la Metodología RUP, para ello se emplea el Modelo del Dominio, los Requerimientos Funcionales y No Funcionales, el Diagrama de Casos de Uso del Sistema, así como una descripción general del sistema que se propone.

2.2 Modelo del Dominio

Un modelo del dominio captura los tipos más importantes de objetos en el contexto del sistema. Los objetos del dominio representan las "cosas" que existen o los eventos que suceden en el entorno en que trabaja el sistema. Muchos de los objetos del dominio o clases pueden obtenerse de una especificación de requisitos o mediante la entrevista con los expertos del dominio (Ivar Jacobson et al., s.d.).

La modelación del dominio tiene como objetivo fundamental la comprensión y descripción de las clases más importantes en el sistema. Para poder entender el contexto en que se desarrolla el sistema se definen los principales conceptos relacionados con el entorno del problema.

2.2.1 Definición de las Entidades y los Conceptos Principales.

- Tarea: Se refiere a una acción concreta a realizar para obtener un resultado deseado en un intervalo de tiempo planificado.
- Flujo de trabajo: Se describe como flujo de trabajo a la secuencia de acciones, actividades o tareas utilizadas para la ejecución de un proceso, incluyendo el seguimiento del estado de cada una de sus etapas y la aportación de las herramientas necesarias para gestionarlo (Ing. Yoandy Pérez Villazón & Ing. Ramón Paumier Samón, 2009).

- **Recurso:** Se refiere a los medios necesarios para el correcto cumplimiento de una tarea.
- **Indicador:** Se refiere a la medida cuantitativa o cualitativa definida para evaluar el nivel de cumplimiento de las tareas en el proceso de Migración.
- **Etapas:** Se refiere al intervalo de tiempo en que se determina desarrollar un conjunto de tareas con un propósito definido y común. Varias etapas pueden ser planificadas para ser desarrolladas de forma paralela.
- **Fase:** Se refiere al intervalo de tiempo en que se planifica una o varias etapas agrupadas en un propósito definido y común.
- **Persona:** Se refiere a los individuos de interés al proceso de migración.
- **Trabajador:** Se refiere a las personas que están vinculadas a la realización de una tarea además del responsable de la misma.
- **Responsable:** Se refiere a aquella persona quien responde por el desarrollo de una parte del Proceso de Migración, trátase de una Fase, Etapa o Tarea.
- **Administrador:** Se refiere a la persona encargada de elaborar y definir la estructura del Proceso de Migración, velando, además, por el correcto desarrollo del mismo.
- **Ordenador:** Se refiere a una computadora, o sea, una máquina electrónica la cual recibe y procesa datos para convertirlos en información útil y que para esto se vale de una gran variedad de secuencias o rutinas de instrucciones llamadas programas.
- **Diagnóstico:** Se refiere al proceso donde se identifica la naturaleza, libre o privativa, de uno o varios programas instalados en un ordenador.
- **Software:** Es un programa, o sea, un conjunto de instrucciones y reglas informáticas para ejecutar ciertas tareas en una computadora (“Real Academia Española,” 2009).
- **Sistema Operativo:** El sistema operativo es el software más importante de un ordenador. Es necesario para que funcionen los otros programas. Los sistemas operativos realizan tareas básicas, tales como enviar la información a la pantalla, tratamiento de archivos y directorios en el disco,

así como el control de dispositivos periféricos (“Definición de Sistema Operativo,” 2009).

- Software de Aplicación: Se refiere a aquel software que se ejecuta a través del Sistema Operativo y que está diseñado para facilitar al usuario la realización de una determinada labor.

2.2.2 Reglas del Negocio a Considerar.

- Las fases se conciben de forma consecutiva en el tiempo de modo que dos o más fases no pueden ser desarrolladas paralelamente.
- Las etapas pueden ser desarrolladas de forma paralela en el tiempo.
- El administrador es el único con permiso para la realización de diagnósticos a ordenadores.
- El administrador tiene permiso para gestionar (añadir, modificar o eliminar) las fases, etapas, tareas, indicadores, recursos, flujos y personas en el sistema.
- El responsable de fase tiene permiso de gestionar (añadir, modificar o eliminar) únicamente las tareas planificadas en su fase.
- El responsable de etapa tiene permiso de gestionar (añadir, modificar o eliminar) únicamente las tareas planificadas en su etapa.
- El responsable de tarea tiene permiso para ver el progreso de su tarea así como para ver y modificar los valores de los respectivos indicadores.
- Todos los usuarios pueden visualizar los reportes y diagramas generales del Proceso de Migración.

2.2.4 Representación del Modelo del Dominio.

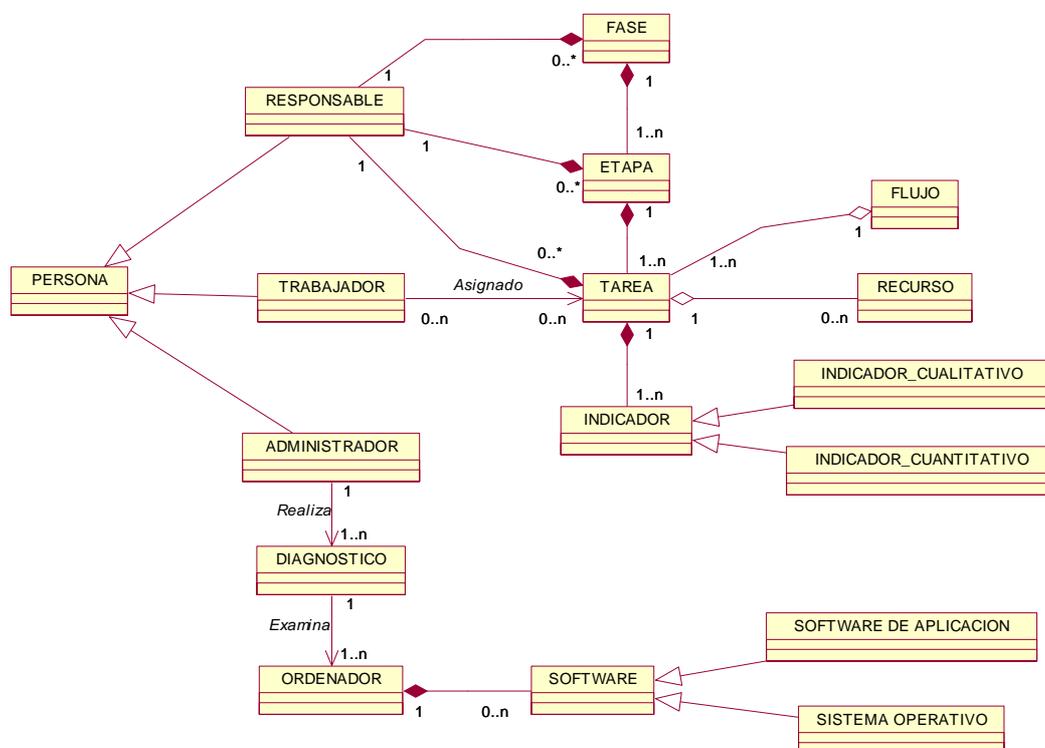


Figura 2.1 Modelo del dominio

2.3 Descripción del Sistema Propuesto.

El manejo de la información hoy en día ha tomado un papel relevante en el desempeño de cualquier empresa, es necesario contar con las herramientas adecuadas y óptimas para almacenar y acceder lo más rápido posible a cualquier tipo de información que precisemos.

Con el sistema propuesto se pretende informatizar la Gestión y Control de la Migración a Software Libre en la Universidad de Cienfuegos “Carlos Rafael Rodríguez”. Su mismo nombre, Sistema de Apoyo a la Migración (SAM), descubre el propósito de esta aplicación Web diseñada para agilizar y organizar este proceso.

SAM está concebido en dos módulos con características propias.

- Gestión: Es donde se manipula y actualiza la información de la migración en lo referente al proceso y su estructura.

- Diagnóstico: Es donde se comprueba la existencia de software libre en los ordenadores de la red, ya sea como sistema operativo o software de aplicación.

Aunque SAM se ajusta a las particularidades de nuestra Universidad, presenta características que lo hacen flexible para su implantación en otros centros donde las estrategias sean diferentes.

El acceso al sistema se efectúa mediante la red utilizando los navegadores Mozilla Firefox o Internet Explorer y luego de una previa autenticación.

Los responsables de Fases y Etapas pueden planificar y controlar correctamente las tareas que se deben acometer. La información es almacenada en la base de datos del sistema cuyo acceso puede ser efectuado en correspondencia con el respectivo permiso. De esta manera es controlado el progreso y el cumplimiento o no de las tareas por parte del personal asignado a las mismas y, de forma general, el desarrollo de la migración.

Este sistema también es capaz de generar e imprimir reportes relacionados con diferentes informaciones del proceso, permite la elaboración de salvadas de seguridad de los datos almacenados, cuenta con una documentada ayuda y un correcto manejo de excepciones. Su acceso puede ser efectuado desde cualquier lugar en la red.

El Sistema de Apoyo a la Migración (SAM) cumple con los siguientes requerimientos:

2.3.1 Requerimientos Funcionales.

- R1. Insertar Personal.
- R2. Modificar Personal.
- R3. Eliminar Personal.
- R4. Listar Personal.
- R5. Insertar Fase.
- R6. Modificar Fase.
- R7. Eliminar Fase.
- R8. Listar Fases.
- R9. Insertar Etapa.
- R10. Modificar Etapa.
- R11. Eliminar Etapa.
- R12. Listar Etapas.
- R13. Insertar Tarea.
- R14. Modificar Tarea.
- R15. Eliminar Tarea.
- R16. Listar Tareas.
- R17. Insertar Indicador.
- R18. Modificar Indicador.
- R19. Eliminar Indicador.
- R20. Listar Indicadores.
- R21. Insertar Permiso.
- R22. Modificar Permiso.
- R23. Eliminar Permiso.
- R24. Listar Permisos.
- R25. Insertar Flujo
- R26. Eliminar Flujo

- R27. Listar Flujos.
- R28. Vincular Trabajador a una Tarea.
- R29. Desvincular Trabajador a una Tarea.
- R30. Listar Trabajadores Vinculados.
- R31. Agregar Recurso a una Tarea.
- R32. Quitar Recurso a una Tarea.
- R33. Listar Recursos Agregados.
- R34. Recuperar Contraseña.
- R35. Enviar Correo con Nueva Contraseña.
- R36. Cambiar Contraseña.
- R37. Mostrar Reporte de Fase.
- R38. Mostrar Reporte de Etapa.
- R39. Mostrar Reporte de Tarea.
- R40. Mostrar Diagrama General de Migración.
- R41. Mostrar Diagnóstico a Equipo con Windows.
- R42. Generar PDF del Reporte de Fase.
- R43. Generar PDF del Reporte de Etapa.
- R44. Generar PDF del Reporte de Tarea.
- R45. Generar PDF del Reporte de Diagrama General de Migración.
- R46. Diagnosticar Red.
- R47. Guardar Datos del Diagnóstico de Red
- R48. Diagnosticar Equipo con Windows
- R49. Descargar Aplicación para Diagnosticar Equipo con Windows
- R50. Guardar Resultado del Diagnóstico a Equipo con Windows.
- R51. Cargar por el Sistema los Resultados del Diagnóstico a Equipo con Windows.
- R52. Listar Diagnósticos a Equipos con Windows realizados con anterioridad.
- R53. Eliminar Diagnóstico a Equipo con Windows que sea innecesario.
- R54. Realizar Salva de Seguridad a la Base de Datos del Sistema.

- R55. Realizar Salva de Seguridad a la Base de Datos de los Diagnósticos Realizados.
- R56. Ver Responsabilidades en la Línea de Tiempo
- R57. Consultar Ayuda.
- R58. Autenticar.

2.3.3 Requerimientos no Funcionales

Los requerimientos no funcionales especifican cualidades, propiedades del sistema; como restricciones del entorno o de la implementación, rendimiento, dependencias de la plataforma, etc (Ivar Jacobson et al., s.d.).

APARIENCIA O INTERFAZ EXTERNA.

- En el caso de la utilidad para el diagnóstico de equipos con Windows, la interfaz debe ser agradable, sencilla y funcional.
- La interfaz del sistema de gestión y diagnóstico debe ser a través de una página Web dinámica que muestre la información de acuerdo a las responsabilidades del usuario que accede.
- La interfaz debe ser intuitiva y amigable que garantice en todo momento al usuario una adecuada orientación en la aplicación así como el control de las funcionalidades.
- La interfaz debe estar en correspondencia con la funcionalidad.
- La interfaz debe garantizar tal sencillez que permita al usuario desarrollar las funcionalidades sin necesidad de bastos conocimientos en el trabajo con computadoras.

USABILIDAD.

- La explotación de este sistema agilizará de forma notable la gestión de la información en el Proceso de Migración a Software Libre.

- El sistema contará con una política de usuarios que impedirá accesos no autorizados que pudieran introducir errores en la información.
- La utilidad de diagnóstico a equipo con Windows así como el diagnóstico a redes solo estarán disponibles para el administrador del sistema.

RENDIMIENTO.

- El sistema debe permitir el acceso simultáneo de los usuarios al sistema.
- El sistema debe recuperarse en un corto período de tiempo ante cualquier falla.

SOPORTE.

- El administrador tendrá bajo su responsabilidad, instalar y mantener la aplicación,
- El sistema debe propiciar su mejoramiento y la inclusión de nuevos módulos en el futuro.

PORTABILIDAD.

- Dadas las características y el propósito de este sistema, el mismo puede ser utilizado en plataforma GNU/Linux así como en Windows. El hecho de ser concebido como aplicación Web contribuye garantiza esta posibilidad.
- La utilidad para diagnosticar equipos con Windows corre únicamente sobre este sistema operativo. Para ser utilizada en GNU/Linux debe utilizarse un emulador como es el caso del WINE (Windows Emulator)

SOFTWARE.

- Se debe disponer de un sistema operativo que soporte Apache como servidor Web y PHP como lenguaje de programación del lado del servidor.
- Como sistema gestor de base de datos se debe utilizar MySQL.

- El sistema está optimizado para ser usado con los navegadores Web Mozilla Firefox e Internet Explorer.

HARDWARE.

- Las máquinas deben estar conectadas a la red de la institución.
- Para las máquinas clientes se requiere un mínimo de 128 MB de RAM.
- En el caso del servidor o servidores para el sistema y la base de datos se requiere un mínimo de 256 MB de RAM (512 MB recomendado).

REQUISITOS LEGALES

- Este producto estará fuera del alcance de los fines comerciales pues su diseño e implementación fueron diseñados con las características de software libre.

SEGURIDAD

- El acceso a la información es restringido al nivel de responsabilidad del que accede por lo que se requiere autenticación. El administrador del sistema es quien establece las responsabilidades en el sistema.
- Las contraseñas son encriptadas mediante el algoritmo MD5.
- La integridad de los datos es fundamental en la política de seguridad del sistema propuesto y para esto contará con un grupo importante de validaciones que no permitan la entrada de datos irreales.
- El sistema garantizará que la información siempre esté disponible, por esta razón se realizarán salvadas de respaldo a la base de datos.

2.4 Modelo de Casos de Uso del Sistema.

Mediante la modelación de los casos de uso es posible llegar a un acuerdo entre los desarrolladores del software y los clientes en cuanto a los requisitos del sistema, o sea, sobre las condiciones y posibilidades que debe cumplir. Describe que hace el sistema para cada tipo de usuario (Ivar Jacobson et al., s.d.).

2.4.1 Actores del Sistema

Un actor no es más que un conjunto de roles que los usuarios de Casos de Uso desempeñan cuando interactúan con estos Casos de Uso. Los actores representan a terceros fuera del sistema que colaboran con el mismo. Una vez que hemos identificado los actores del sistema, tenemos identificado el entorno externo del sistema (Ivar Jacobson et al., s.d.).

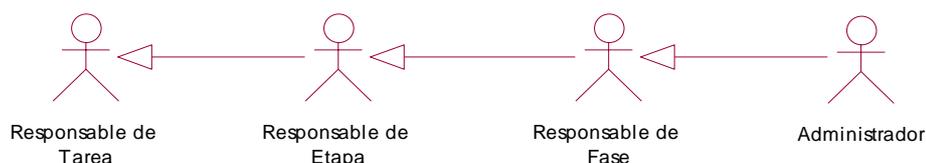


Figura 2.2 Actores del sistema.

Actor	Justificación
Responsable de Tarea	Vela por el cumplimiento de la(s) tarea(s) asignada(s) a su cargo. Mantiene actualizada la información correspondiente al progreso de las mismas. Accede a la información de interés personal y General del Proceso de Migración
Responsable de Etapa	Es una especialización del Responsable de Tareas y además vela por el cumplimiento de la(s) etapa(s) asignada(s) a su cargo. Concibe tanto las tareas a emprender como sus indicadores, vincula

	trabajadores y asigna recursos a las tareas.
Responsable de Fase	Es una especialización del Responsable de Etapa y además vela por el cumplimiento y desarrollo de la(s) fase(s) asignada(s) a su cargo e implícitamente las etapas enmarcadas en las mismas.
Administrador	Es una especialización del Responsable de Fase pero además: Es quien configura la estructura general de la Estrategia de Migración definiendo las Fases y Etapas necesarias. Define los flujos de trabajo. Es quien otorga los permisos al sistema, realiza los diagnósticos a redes y equipos así como las salvadas de seguridad del sistema.

Tabla 2.1 Actores del sistema.

2.4.2 Paquetes y sus Relaciones.

Un caso de uso especifica una secuencia de acciones que el sistema puede llevar a cabo interactuando con sus actores, incluyendo alternativas dentro de la secuencia (Ivar Jacobson et al., s.d.).

Para lograr una mayor claridad en la representación de los casos de uso, estos fueron agrupados en paquetes de acuerdo a las funcionalidades del sistema.

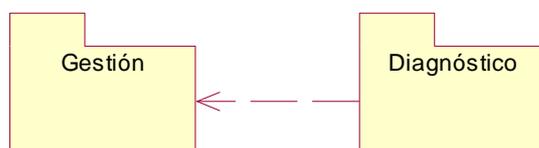


Figura 2.3 Paquetes

Gestión:

- Autenticar.

- Recuperar Contraseña.
- Cambiar Contraseña.
- Gestionar Indicador.
- Gestionar Tarea.
- Gestionar Vinculación de Trabajadores.
- Gestionar Asignación de Recursos.
- Visualizar Fase.
- Gestionar Etapa.
- Gestionar Fase.
- Gestionar Personal.
- Gestionar Flujo.
- Gestionar Permiso.
- Generar Reporte Exportable.
- Visualizar Responsabilidades.
- Modificar Indicador.
- Realizar Salva de Seguridad al Sistema de Gestión.
- Consultar la Ayuda.

Diagnóstico:

- Diagnosticar Equipo.
- Descargar Utilidad de Diagnóstico de Equipo.
- Subir Diagnóstico Realizado a Equipo.
- Eliminar Diagnóstico Innecesario.
- Visualizar Diagnóstico Realizado a Equipo.
- Diagnosticar Red.
- Realizar Salva de Seguridad al Registro de Diagnósticos.

2.4.3 Diagramas de Casos de Uso del Sistema por Paquetes.

Gestión:

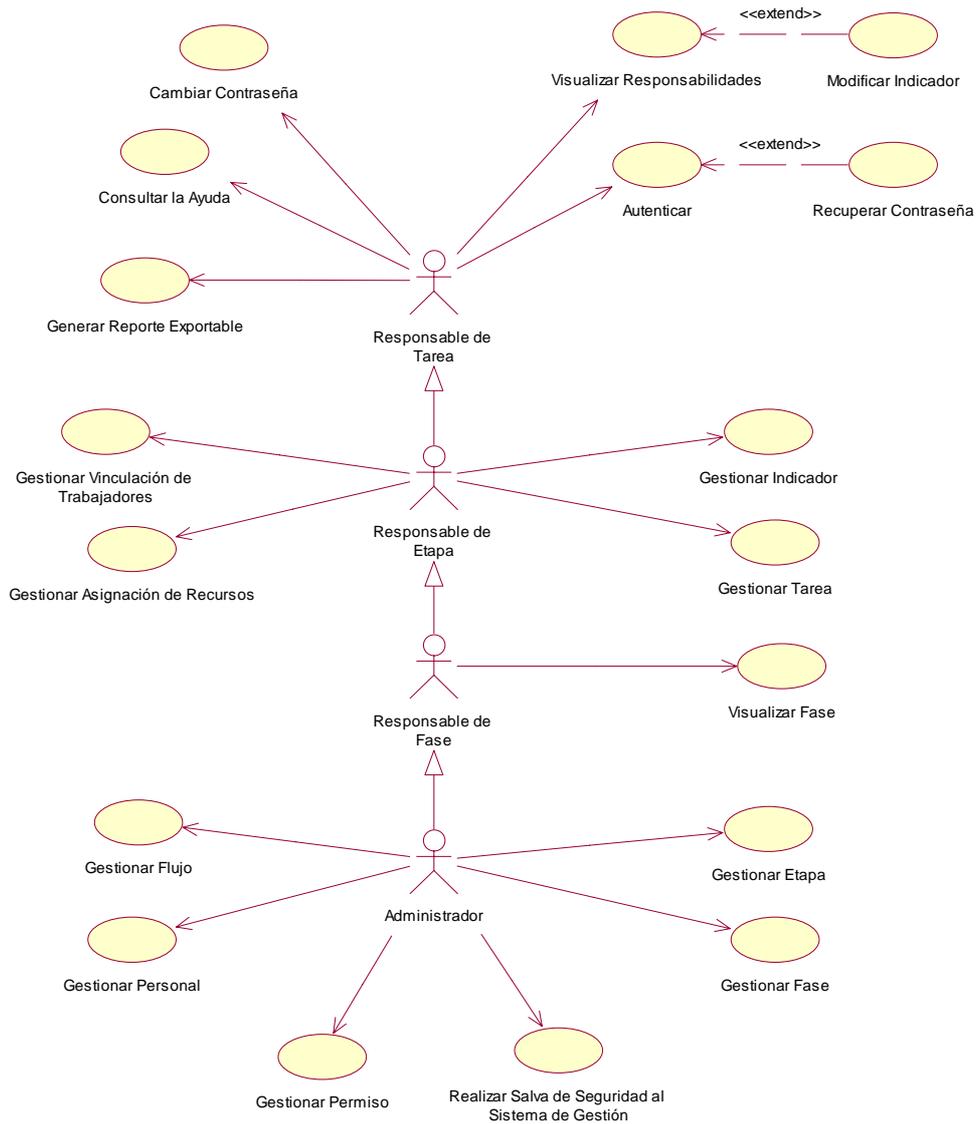


Figura 2.4 Casos de uso del paquete de gestión.

Diagnóstico:

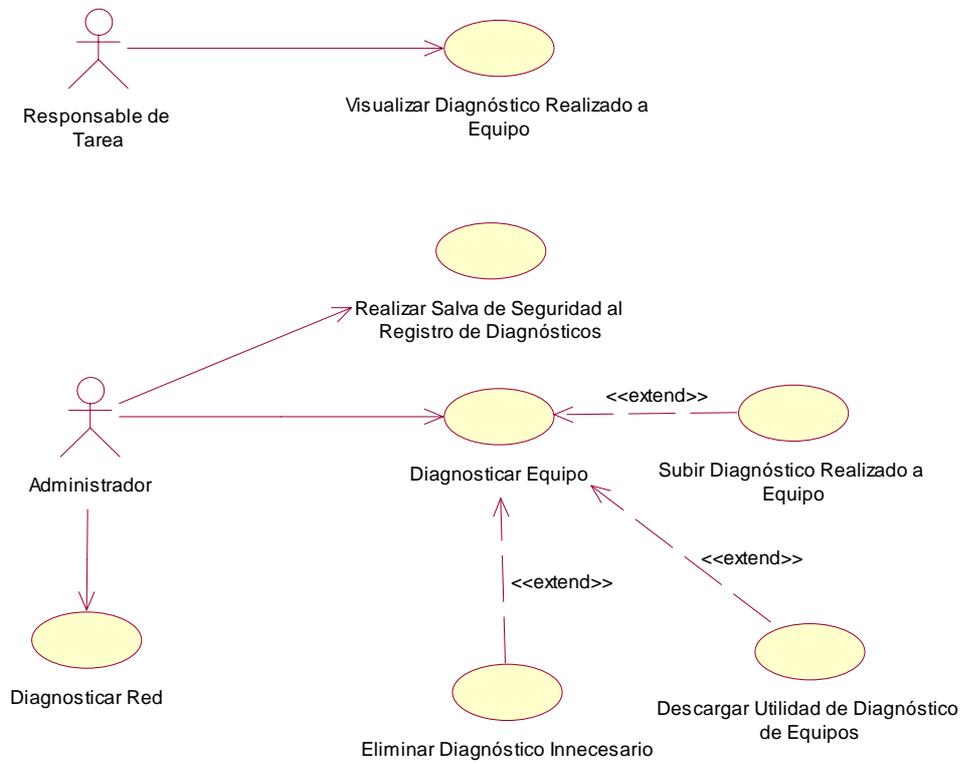


Figura 2.5 Casos de uso del paquete de diagnóstico.

2.4.4 Descripción de los Casos de Uso del Sistema.

Gestión:

Caso de uso	Autenticar
Actores	Responsable de Tarea (inicia)
Propósito	Garantizar el acceso al software solo a los usuarios autorizados.
Resumen	
<p>El caso de uso comienza cuando el Responsable de Tarea accede a la aplicación Web, introduciendo su nombre de usuario y contraseña, el sistema se comunica con la base de datos para comprobar que realmente es un usuario del sistema, si el intento de ingreso resulta denegado el usuario tiene la posibilidad de volver a intentar su ingreso. El caso de uso termina cuando la autenticación se realiza con éxito.</p>	
Referencias	R59, CU Recuperar Contraseña
Precondiciones	---
Post-condiciones	El usuario del sistema tiene garantizado el acceso a las funcionalidades del software que le corresponden.
Prototipo	ANEXO 2

Tabla 2.2 Caso de uso: Autenticar.

Caso de uso	Recuperar Contraseña
Actores	Responsable de Tarea (inicia)
Propósito	Garantizar la recuperación de contraseña de acceso al sistema de forma segura.
Resumen	
<p>El caso de uso inicia cuando el Responsable de Tarea accede a la aplicación Web habiendo olvidado su contraseña, entonces selecciona la opción para recuperar contraseña. Introduciendo su nombre de usuario se le realiza una pregunta de</p>	

seguridad previamente definida. Si contesta correctamente la pregunta, se genera una nueva contraseña aleatoria la cual le es enviada por correo electrónico para su inmediato cambio. El caso de uso termina cuando el responsable de tarea recibe en su correo la nueva contraseña para ser inmediatamente cambiada.	
Referencias	R34, R35, R36
Precondiciones	La pregunta de seguridad y el correo deben haber sido definidos previamente.
Post-condiciones	La contraseña ha sido cambiada satisfactoriamente y enviada al correo del responsable de tareas.
Prototipo	ANEXO 3

Tabla 2.3 Caso de uso: Recuperar Contraseña.

Caso de uso	Cambiar Contraseña
Actores	Responsable de Tarea (inicia)
Propósito	Garantizar el cambio de la contraseña de acceso al sistema.
Resumen	
El caso de uso comienza cuando el responsable de tarea selecciona la opción cambiar contraseña. En este caso se introduce la contraseña anterior como medida de seguridad y a continuación la nueva contraseña. Si lo desea además puede modificar la pregunta de seguridad. El caso de uso termina cuando la contraseña ha sido cambiada satisfactoriamente.	
Referencias	R36
Precondiciones	El responsable de tarea debe de tener previamente asignado un permiso de acceso al sistema.
Post-condiciones	El permiso para este usuario queda correctamente actualizado.
Prototipo	ANEXO 4

Tabla 2.4 Caso de uso: Cambiar Contraseña.

Caso de uso	Gestionar Indicador
Actores	Responsable de Etapa (inicia)

Propósito	Establecer los indicadores por los que se medirá el progreso de determinada tarea.
Resumen	
<p>El caso de uso se inicia cuando se requiere registrar un nuevo indicador a determinada tarea o se necesita modificar uno de los indicadores existentes. En estos casos, el responsable de etapa registra la información del indicador o los cambios que se produzcan en sus datos. El caso de uso finaliza con el registro de los indicadores en esta tarea correctamente actualizado. Si se va a eliminar un indicador se muestra una advertencia antes de proceder, de estar de acuerdo, se procede a su eliminación.</p>	
Referencias	R17, R18, R19 y R20
Precondiciones	En caso de modificar o eliminar es necesario que el registro haya sido previamente insertado.
Post-condiciones	El listado de indicadores de esta tarea queda actualizado.
Prototipo	ANEXO 5

Tabla 2.5 Caso de uso: Gestionar Indicador.

Caso de uso	Gestionar Tarea
Actores	Responsable de Etapa (inicia)
Propósito	Establecer las tareas a realizar en determinada etapa.
Resumen	
<p>El caso de uso se inicia cuando se requiere registrar una nueva tarea en el una etapa o se necesita modificar uno de los registros de tarea existentes. En estos casos, el responsable de etapa registra la información de la tarea o los cambios que se produzcan en sus datos. El caso de uso finaliza con el registro de la tarea actualizado. Si se va a eliminar una tarea se muestra una advertencia antes de proceder, de estar de acuerdo, se procede a su eliminación.</p>	
Referencias	R13, R14, R15 y R16
Precondiciones	En caso de modificar o eliminar es necesario que el registro

	haya sido previamente insertado.
Post-condiciones	El listado de tareas de esta etapa queda actualizado.
Prototipo	ANEXO 6

Tabla 2.6 Caso de uso: Gestionar Tarea.

Caso de uso	Gestionar Vinculación de Trabajadores
Actores	Responsable de Etapa (inicia)
Propósito	Establecer que trabajadores, además del responsable, están vinculados al cumplimiento de una tarea.
Resumen	
<p>El caso de uso se inicia cuando se requiere actualizar el listado de trabajadores vinculados a una determinada tarea. En este caso, el responsable de etapa vincula o desvincula trabajadores a una tarea a partir del listado de personas registradas en el sistema. El caso de uso finaliza con el listado de trabajadores actualizado en la tarea.</p>	
Referencias	R28, R29, R30
Precondiciones	Los trabajadores deben tener previamente sus datos registrados en el listado de personal.
Post-condiciones	El listado de trabajadores vinculados a esta tarea queda actualizado.
Prototipo	ANEXO 7

Tabla 2.7 Caso de uso: Gestionar Vinculación de Trabajadores.

Caso de uso	Gestionar Asignación de Recursos
Actores	Responsable de Etapa (inicia)
Propósito	Registrar los recursos asignados a una determinada tarea para lograr su cumplimiento.
Resumen	
<p>El caso de uso inicia cuando se hace necesario actualizar el registro de recursos asignados a una tarea. En este caso, el responsable de etapa inserta o elimina</p>	

recursos a la tarea en cuestión. El caso de uso finaliza con el listado de recursos asignados correctamente actualizado.	
Referencias	R31, R32, R33
Precondiciones	La tarea en cuestión debe estar previamente definida.
Post-condiciones	El listado de recursos asignados a esta tarea queda actualizado.
Prototipo	ANEXO 8

Tabla 2.8 Caso de uso: Gestionar Asignación de Recursos.

Caso de uso	Visualizar Fase
Actores	Responsable de Fase (inicia)
Propósito	Mostrar la información de las fases a sus responsables.
Resumen	
El caso de uso se inicia cuando un responsable de fase selecciona la opción visualizar los datos en una de las fases en que es responsable. En este caso se le muestra la información correspondiente.	
Referencias	R8
Precondiciones	El responsable debe tener a su cargo al menos una fase.
Post-condiciones	---
Prototipo	ANEXO 9

Tabla 2.9 Caso de uso: Visualizar Fase.

Caso de uso	Gestionar Etapa
Actores	Administrador (inicia)
Propósito	Establecer las etapas correspondientes a cada fase en la Estructura de la Migración.
Resumen	
El caso de uso se inicia cuando se requiere registrar una nueva etapa a una fase o se necesita modificar uno de los registros de etapa existentes. En estos casos, el administrador registra la información de la etapa o los cambios que se produzcan	

<p>en sus datos. El caso de uso finaliza con el registro de la etapa actualizado. Si se necesita eliminar una etapa se muestra una advertencia antes de proceder, de estar de acuerdo, se procede a su eliminación incluyendo las tareas contenidas en ella.</p>	
Referencias	R9, R10, R11, R12
Precondiciones	En caso de modificar o eliminar es necesario que el registro haya sido previamente insertado.
Post-condiciones	El listado de etapas queda actualizado.
Prototipo	ANEXO 10

Tabla 2.10 Caso de uso: Gestionar Etapa.

Caso de uso	Gestionar Fase
Actores	Administrador (inicia)
Propósito	Establecer las fases para la Estructura de la Migración.
Resumen	
<p>El caso de uso se inicia cuando se requiere registrar una nueva fase en el sistema o se necesita modificar uno de los registros de fase existentes. En estos casos, el administrador registra la información de la fase o los cambios que se produzcan en sus datos. El caso de uso finaliza con el registro de la fase actualizado. Si se va a eliminar una fase se muestra una advertencia antes de proceder, de estar de acuerdo, se procede a eliminar la fase, sus etapas y sus tareas.</p>	
Referencias	R5, R6, R7 y R8
Precondiciones	En caso de modificar o eliminar es necesario que el registro haya sido previamente insertado.
Post-condiciones	El listado de fases queda actualizado.
Prototipo	ANEXO 11

Tabla 2.11 Caso de uso: Gestionar Fase.

Caso de uso	Gestionar Personal
Actores	Administrador (inicia)
Propósito	Actualizar la información del personal involucrado en la

	Migración.
Resumen	
<p>El caso de uso se inicia cuando se requiere registrar una nueva persona en el sistema o se necesita modificar uno de los registros existentes. En estos casos, el administrador registra la información de la persona o los cambios que se produzcan en sus datos. El caso de uso finaliza con el registro de la persona actualizado. Si se va a eliminar una persona, se verifica si tiene alguna responsabilidad en Fase, Etapa o Tarea, en cuyo caso no se realiza la eliminación; de otro modo se procede a la eliminación física.</p>	
Referencias	R1, R2, R3 y R4
Precondiciones	En caso de modificar o eliminar es necesario que el registro haya sido previamente insertado.
Post-condiciones	El listado del personal queda actualizado.
Prototipo	ANEXO 12

Tabla 2.12 Caso de uso: Gestionar Personal.

Caso de uso	Gestionar Flujo
Actores	Administrador (inicia)
Propósito	Establecer los flujos de trabajo en que se enmarcarán las tareas.
Resumen	
<p>El caso de uso se inicia cuando se requiere establecer los flujos de trabajo en que se desarrollarán las tareas. En este caso, el administrador procede a definir el flujo de trabajo. Si se va a eliminar un flujo se muestra una advertencia antes de proceder, de estar de acuerdo, se procede a su eliminación. El caso de uso culmina con la lista de flujos de trabajo correctamente actualizada.</p>	
Referencias	R25, R26 y R27
Precondiciones	En el caso de eliminar, debe haber al menos un flujo establecido.

Post-condiciones	El registro de flujos en el sistema queda correctamente actualizado.
Prototipo	ANEXO 13

Tabla 2.13 Caso de uso: Gestionar Flujo.

Caso de uso	Gestionar Permiso
Actores	Administrador (inicia)
Propósito	Proteger la información del sistema a través de la administración del personal con acceso.
Resumen	
<p>El caso de uso se inicia cuando se requiere conceder un permiso de acceso al sistema o modificar un permiso existente. En estos casos, el administrador seleccionará del personal ya registrado, aquel a quien desea crear o modificar el permiso. El administrador registra la información del permiso o los cambios que se produzcan en sus datos. El caso de uso finaliza con el registro del permiso correctamente actualizado. Si se va a eliminar un permiso se muestra una advertencia antes de proceder, de estar de acuerdo, se procede a su eliminación.</p>	
Referencias	R21, R22, R23 y R24
Precondiciones	La persona a quien se desea conceder el permiso debe estar previamente registrada.
Post-condiciones	El registro de permisos en el sistema queda correctamente actualizado.
Prototipo	ANEXO 14

Tabla 2.14 Caso de uso: Gestionar Permiso.

Caso de uso	Generar Reporte Exportable
Actores	Responsable de Tarea (inicia)
Propósito	Generar reportes útiles para describir la estructura y el progreso de la migración.
Resumen	

El caso de uso inicia cuando se requiere visualizar uno de los reportes exportables generados por el sistema. En este caso se selecciona el tipo de reporte: (de Fase, Etapa, Tarea o Diagrama General de la Migración) y se procede a su generación. De ser necesario el responsable de tarea puede exportar el reporte como un fichero PDF. El Caso de uso culmina con el reporte correctamente generado.	
Referencias	R37, R38, R39, R40, R42, R43, R44, R45
Precondiciones	La información necesaria para generar el reporte solicitado debe existir en el sistema.
Post-condiciones	El reporte es generado satisfactoriamente.
Prototipo	ANEXO 15

Tabla 2.15 Caso de uso: Generar Reporte Exportable.

Caso de uso	Visualizar Responsabilidades
Actores	Responsable de Tarea (inicia)
Propósito	Mantener actualizado el progreso de las tareas.
Resumen	
El caso de uso comienza cuando el responsable de tarea necesita chequear las tareas cuyas responsabilidades le han sido delegadas para de esta forma mantener actualizada la información sobre el progreso de las mismas. Puede escoger entre visualizar en forma de listado o visualizar en línea de tiempo. El caso de uso culmina con la información del progreso de las tareas correctamente actualizada.	
Referencias	R16, R56, CU Modificar Indicador
Precondiciones	El usuario actual debe tener a su cargo al menos una tarea.
Post-condiciones	La información sobre el progreso de las tareas correspondientes al responsable actual queda correctamente actualizada.
Prototipo	ANEXO 16

Tabla 2.16 Caso de uso: Visualizar Responsabilidades.

Caso de uso	Modificar Indicador
Actores	Responsable de Tarea
Propósito	Actualizar el valor de un indicador en determinada tarea.
Resumen	
<p>El caso de uso se inicia cuando el responsable de tarea requiere actualizar el valor de un indicador en una determinada tarea. En este caso, se procede a la modificación del valor actual. El caso de uso finaliza con el registro de los indicadores en esta tarea correctamente actualizado.</p>	
Referencias	R18, R20
Precondiciones	La tarea en cuestión debe tener definido al menos un indicador.
Post-condiciones	El listado de indicadores de esta tarea queda actualizado.
Prototipo	ANEXO 17

Tabla 2.17 Caso de uso: Modificar Indicador.

Caso de uso	Realizar Salva de Seguridad al Sistema de Gestión.
Actores	Administrador (inicia)
Propósito	Garantizar un respaldo de seguridad a la información relacionada con la estructura de la migración y el funcionamiento del sistema de gestión.
Resumen	
<p>El caso de uso inicia cuando el administrador selecciona la opción de realizar salva de seguridad al sistema. En este caso se le presenta un cuadro de diálogo para elegir donde desea almacenar el fichero de respaldo. El caso de uso culmina cuando se ha efectuado satisfactoriamente la salva de seguridad a la base de datos de gestión.</p>	
Referencias	R54
Precondiciones	---
Post-condiciones	---

Prototipo	ANEXO 18
------------------	----------

Tabla 2.18 Caso de uso: Realizar Salva de Seguridad al Sistema de Gestión.

Caso de uso	Consultar Ayuda
Actores	Responsable de Tarea (inicia)
Propósito	Proporcionar información sobre el sistema y sus funcionalidades.
Resumen	
<p>El caso de uso inicia cuando el responsable de tarea selecciona la opción de ayuda. En este caso se carga la página de ayuda en otro navegador para consultar las dudas que puedan existir sobre el sistema en cuestión. El caso de uso culmina cuando se ha mostrado de forma satisfactoria la ayuda del sistema.</p>	
Referencias	R56
Precondiciones	---
Post-condiciones	---
Prototipo	ANEXO 19

Tabla 2.19 Caso de uso: Consultar Ayuda.

Diagnostico:

Caso de uso	Diagnosticar Equipo
Actores	Administrador (inicia)
Propósito	Obtener un listado de las aplicaciones instaladas en un equipo remoto con sistema operativo Windows.
Resumen	
<p>El caso de uso inicia cuando el administrador se dispone a diagnosticar un equipo con sistema operativo Windows. En este caso se utiliza la herramienta VerInstalados, de no poseerla es descargada en el vínculo proporcionado para esta funcionalidad. Suministrando el nombre o el IP de la estación de trabajo se generará el listado de aplicaciones instaladas. Este diagnóstico una vez salvado</p>	

<p>es subido al sistema para ser analizado. Los diagnósticos ya innecesarios pueden ser eliminados del sistema. El Caso de uso culmina cuando el listado de diagnósticos realizados está correctamente actualizado con aquella información que pueda resultar de utilidad posteriormente.</p>	
Referencias	R48, R50, CU Descargar Utilidad de Diagnóstico a Equipo, CU Subir Diagnóstico Realizado a Equipo, CU Eliminar Diagnóstico Innecesario
Precondiciones	Es necesario que exista al menos un archivo en el registro de diagnósticos a equipos con Windows.
Post-condiciones	Queda generado un listado de aplicaciones libres conocidas y de otra naturaleza, instaladas en el equipo analizado.
Prototipo	ANEXO 20

Tabla 2.20 Caso de uso: Diagnosticar Equipo.

Caso de uso	Descargar Utilidad de Diagnóstico de Equipo
Actores	Administrador (inicia)
Propósito	Proporcionar la utilidad para el diagnóstico de equipos con Windows.
Resumen	
<p>El caso de uso inicia cuando el administrador se dispone a diagnosticar un equipo con sistema operativo Windows y no posee la herramienta VerInstalados. En este caso, seleccionando el vínculo habilitado para su obtención, la misma es descargada. El Caso de uso culmina cuando la aplicación ver instalados es descargada satisfactoriamente por el administrador.</p>	
Referencias	R49
Precondiciones	---
Post-condiciones	El administrador se encuentra listo para proceder a diagnosticar un equipo con sistema operativo Windows .
Prototipo	ANEXO 21

Tabla 2.21 Caso de uso: Descargar Utilidad de Diagnóstico de Equipo.

Caso de uso	Subir Diagnóstico Realizado a Equipo.
Actores	Administrador (inicia)
Propósito	Almacenar un diagnóstico a equipo con sistema operativo Windows para posterior referencia.
Resumen	
<p>El caso de uso inicia cuando el administrador selecciona la opción analizar los resultados obtenidos por el programa VerInstalados. En este caso procede a examinar la ubicación del fichero generado por el programa y a continuación carga el mismo al registro existente en el sistema. El caso de uso culmina cuando se ha guardado satisfactoriamente el diagnóstico quedando así disponible para posteriores referencias.</p>	
Referencias	R51
Precondiciones	Debe existir un diagnóstico previo realizado mediante la utilidad VerInstalados.
Post-condiciones	El diagnóstico es guardado en un directorio destinado para esto y además a quedado registrado en la base de datos.
Prototipo	ANEXO 22

Tabla 2.22 Caso de uso: Subir Diagnóstico Realizado a Equipo.

Caso de uso	Eliminar Diagnóstico Innecesario.
Actores	Administrador (inicia)
Propósito	Garantizar que el listado de diagnósticos a equipos con Windows no contenga registros innecesarios.
Resumen	
<p>El caso de uso inicia cuando el administrador selecciona la opción eliminar en el listado de diagnósticos guardados. En este caso se muestra una advertencia, si esta de acuerdo se procede a eliminar el registro en la base de datos así como el diagnóstico guardado. El caso de uso culmina con la base de datos y el registro de diagnósticos realizados correctamente actualizados.</p>	

Referencias	R52, R53
Precondiciones	Debe existir al menos un diagnóstico registrado.
Post-condiciones	La base de datos y el registro de diagnósticos están limpios de información innecesaria.
Prototipo	ANEXO 23

Tabla 2.23 Caso de uso: **Eliminar Diagnóstico Innecesario.**

Caso de uso	Visualizar Diagnóstico Realizado a Equipo
Actores	Responsable de Tarea (inicia)
Propósito	Mostrar el resultado de un diagnóstico realizado a un equipo con Windows.
Resumen	
<p>El caso de uso inicia cuando el responsable de tarea selecciona la opción de visualizar análisis previamente realizados. En este caso se le muestra un listado ordenado por fecha de los diagnósticos efectuados en terminales con Windows. Luego de seleccionar el fichero a analizar se presenta un listado de las aplicaciones catalogadas como software libres y seguidamente las demás aplicaciones instaladas no reconocidas en esta categoría. El Caso de uso culmina con el reporte correctamente generado.</p>	
Referencias	R41, R52
Precondiciones	Es necesario que exista al menos un archivo en el registro de diagnósticos a equipos con Windows.
Post-condiciones	Queda generado un listado de aplicaciones libres conocidas y de otra naturaleza, instaladas en el equipo analizado.
Prototipo	ANEXO 24

Tabla 2.24 Caso de uso: **Visualizar Diagnóstico Realizado a Equipo.**

Caso de uso	Diagnosticar Red.
Actores	Administrador (inicia)
Propósito	Obtener un diagnóstico de los sistemas operativos utilizados actualmente en la red.

Resumen	
<p>El caso de uso inicia cuando el administrador selecciona la opción de realizar diagnóstico a la red. En este caso se le presentan tres opciones entre las que deberá escoger qué tipo de diagnóstico realizará:</p> <ul style="list-style-type: none"> - Por rango de IP. - Por IP y máscara. - Por nombre y máscara. <p>Una vez seleccionada la opción procederá a entrar los datos necesarios para el diagnóstico, el sistema comenzará a diagnosticar cada uno de los equipos en el rango especificado. Al terminar se le mostrará un reporte con los sistemas operativos en uso por computadora así como el porcentaje de uso de GNU/Linux y Windows. Estos datos pueden ser salvados en un fichero. El caso de uso culmina cuando se presenta el informe de sistemas operativos uso por el rango de maquinas especificado.</p>	
Referencias	R46, R47
Precondiciones	---
Post-condiciones	---
Prototipo	ANEXO 25

Tabla 2.25 Caso de uso: Diagnosticar Red.

Caso de uso	Realizar Salva de Seguridad al Registro de Diagnósticos.
Actores	Administrador (inicia)
Propósito	Garantizar un respaldo de seguridad a la información relacionada con los diagnósticos almacenados en el sistema.
Resumen	
<p>El caso de uso inicia cuando el administrador selecciona la opción de realizar salva de seguridad al registro de diagnósticos. En este caso se le presenta un cuadro de diálogo para elegir donde desea almacenar el fichero de respaldo. El</p>	

caso de uso culmina cuando se ha efectuado satisfactoriamente la salva de seguridad a la base de datos de diagnósticos.	
Referencias	R55
Precondiciones	---
Post-condiciones	---
Prototipo	ANEXO 18

Tabla 2.26 Caso de uso: Realizar Salva de Seguridad al Registro de Diagnósticos.

2.5 Conclusiones

En este capítulo se ha descrito la solución propuesta utilizando el Modelo del Dominio. Se definieron los conceptos asociados al dominio del problema y sus relaciones. También fueron expuestos los requerimientos funcionales y no funcionales, así como los Casos de Uso con sus respectivas descripciones.

Capítulo 3: Construcción de la solución propuesta

3.1 Introducción.

En este capítulo se describe la construcción de la solución propuesta. Con este fin se utilizan los diagramas de clases Web del diseño así como el modelo lógico y físico de la base de datos. Se describen los principios de diseño utilizados y la concepción de la ayuda. Para describir los elementos fundamentales de la implementación se muestra el diagrama de implementación. Se realiza, además, un cálculo para estimar la factibilidad del proyecto.

3.2 Diagrama de clases del diseño.

El diagrama de clases del diseño representa las colaboraciones que ocurren entre las páginas que conforman la aplicación Web. Cada página lógica puede ser representada como una clase. Este diagrama, además, permite representar el nivel de abstracción adecuado y la relación con los restantes artefactos de UML.

A continuación se presentan los diagramas de clases Web desarrollados a partir de los casos de uso del sistema.

Autenticar

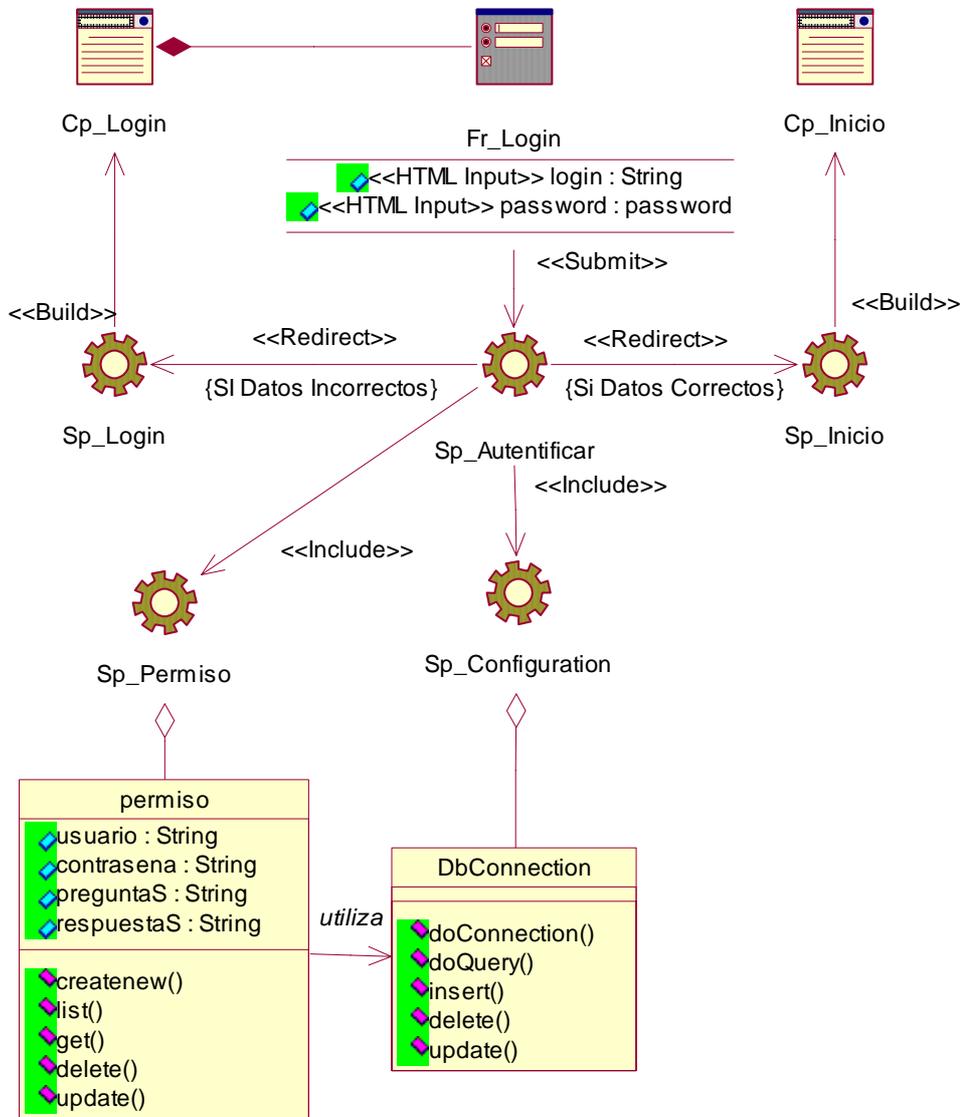


Figura 3.1 Caso de uso: Autenticar.

Recuperar Contraseña

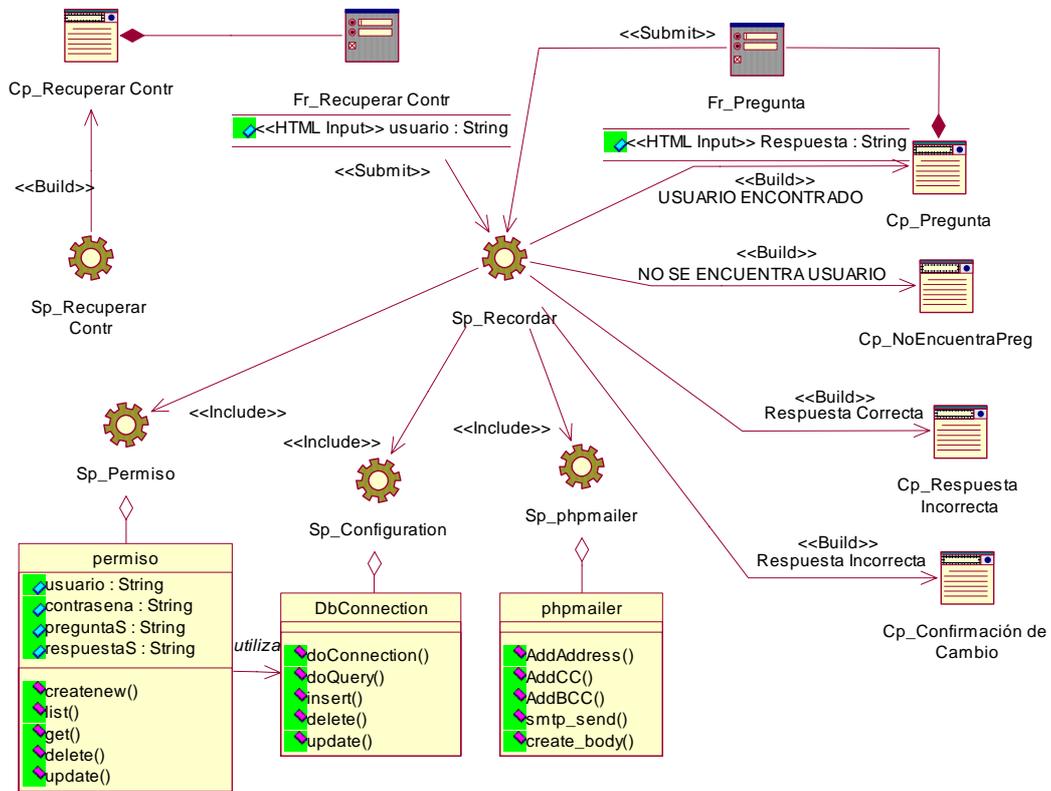


Figura 3.2 Caso de uso: Recuperar Contraseña.

Cambiar Contraseña

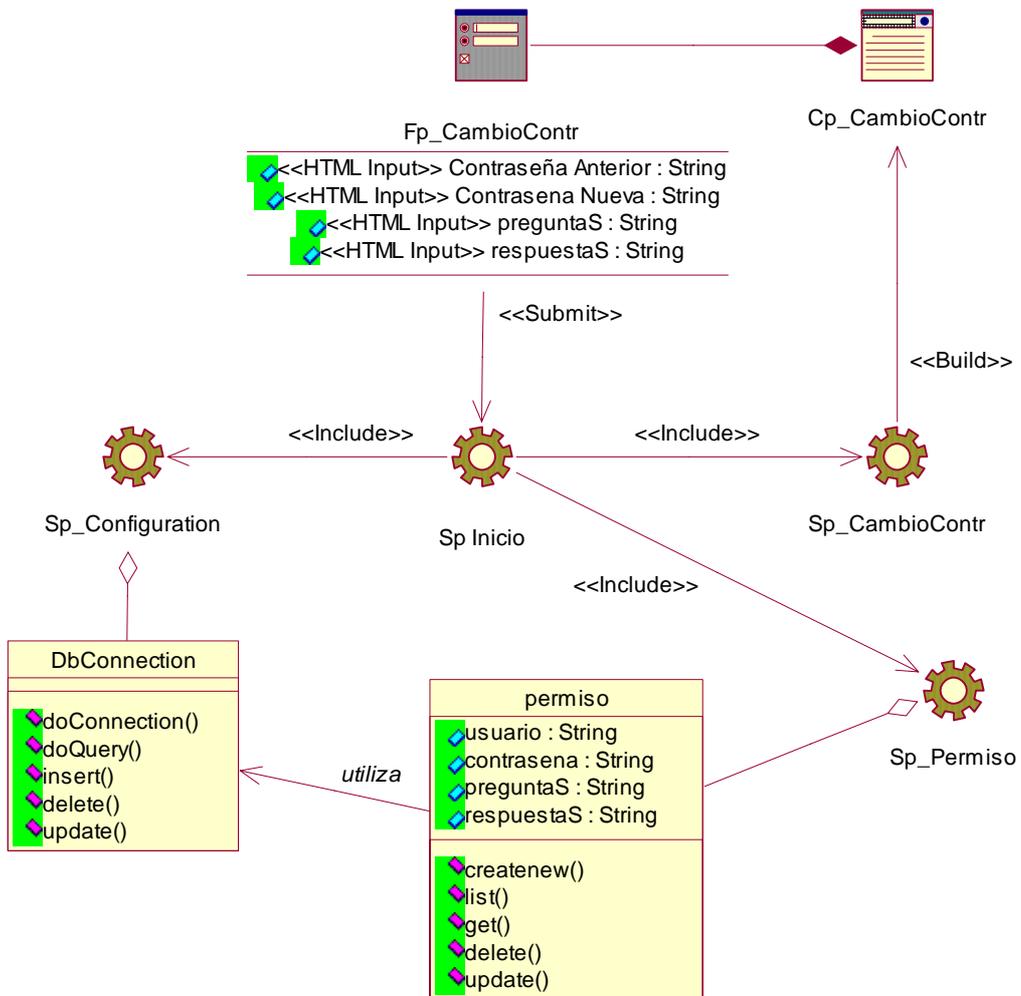


Figura 3.3 Caso de uso: Cambiar Contraseña.

Gestionar Indicador

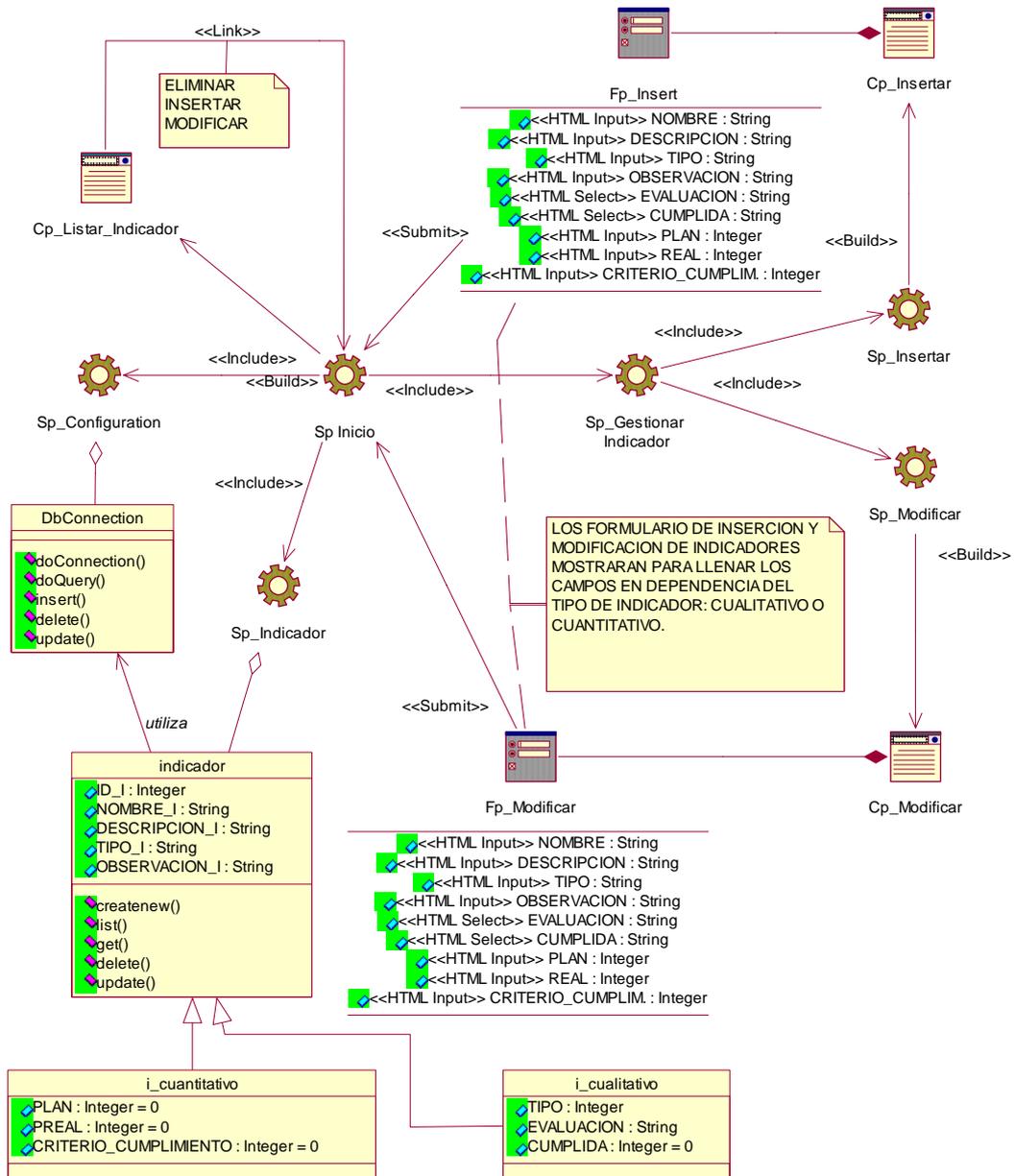


Figura 3.4 Caso de uso: Gestionar Indicador.

Gestionar Tarea

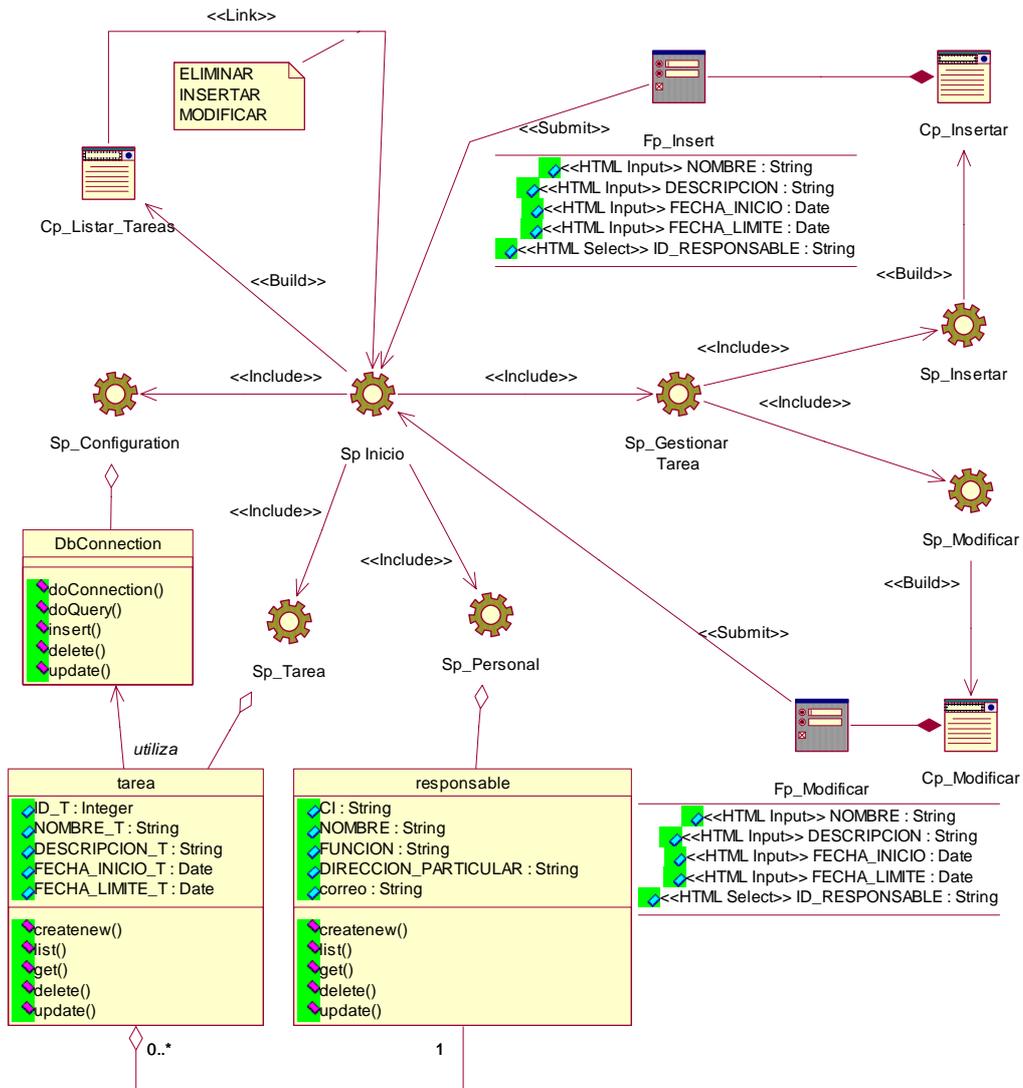


Figura 3.5 Caso de uso: Gestionar Tarea.

Gestionar Vinculación de Trabajadores

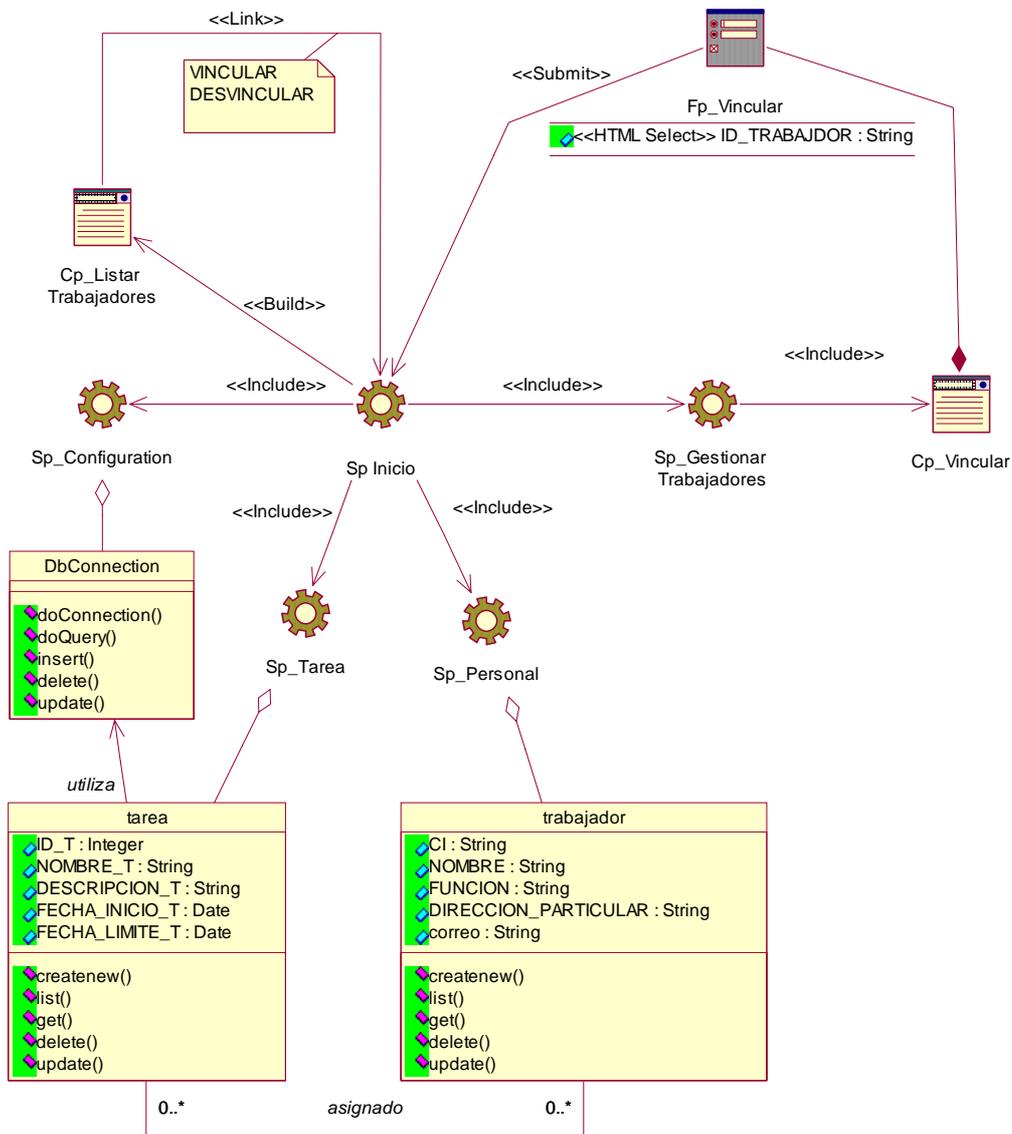


Figura 3.6 Caso de uso: Gestionar Vinculación de Trabajadores.

Gestionar Asignación de Recursos

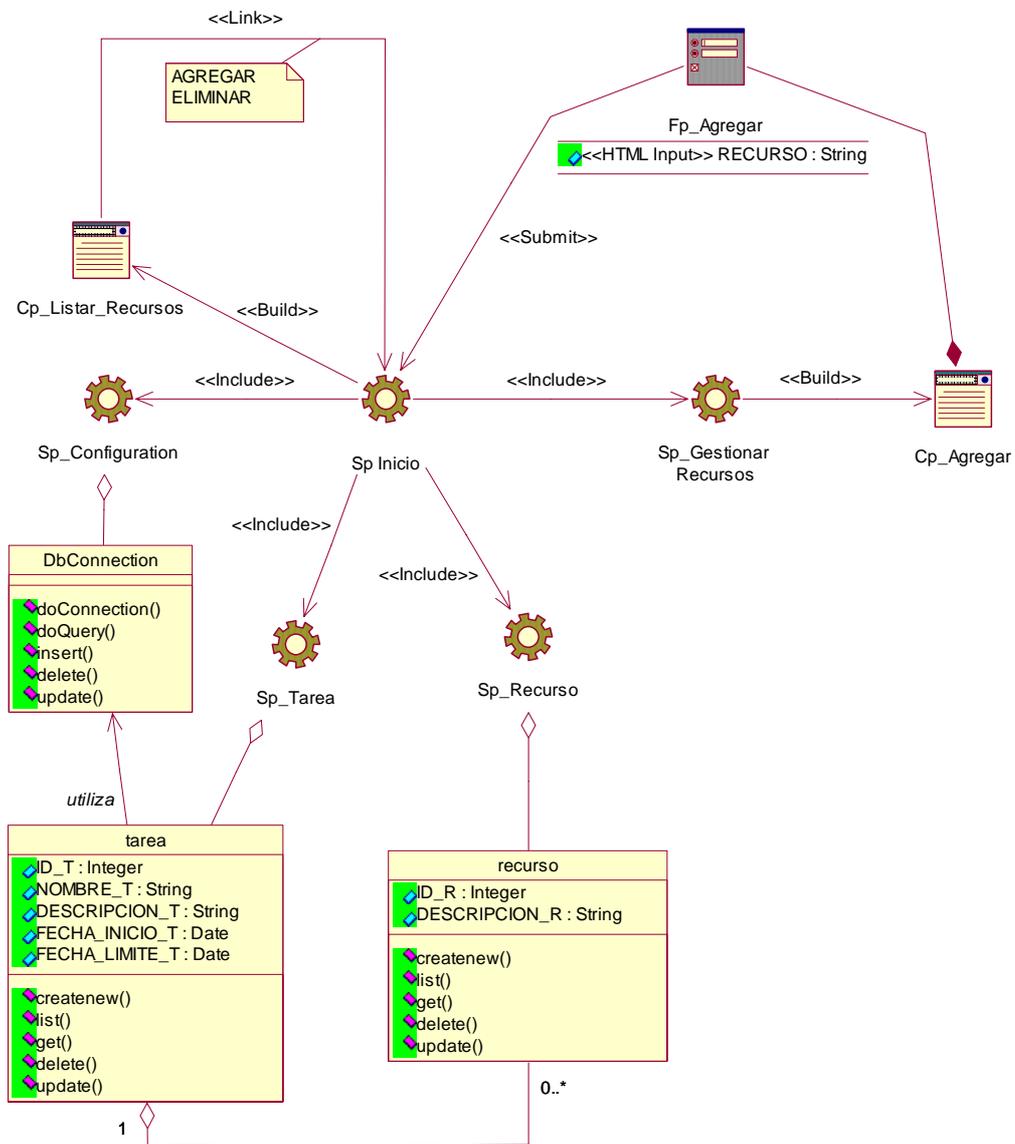


Figura 3.7 Caso de uso: Gestionar Asignación de Recursos.

Visualizar Fase

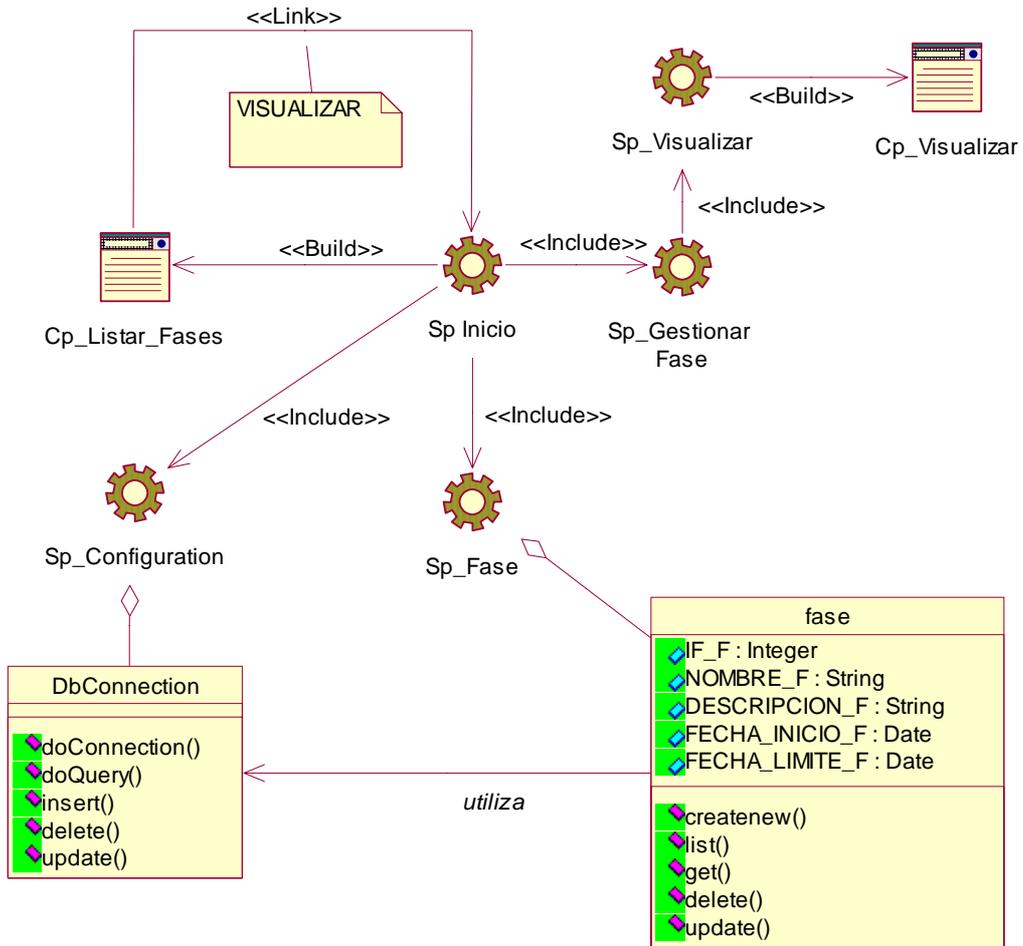


Figura 3.8 Caso de uso: Visualizar Fase.

Gestionar Etapa

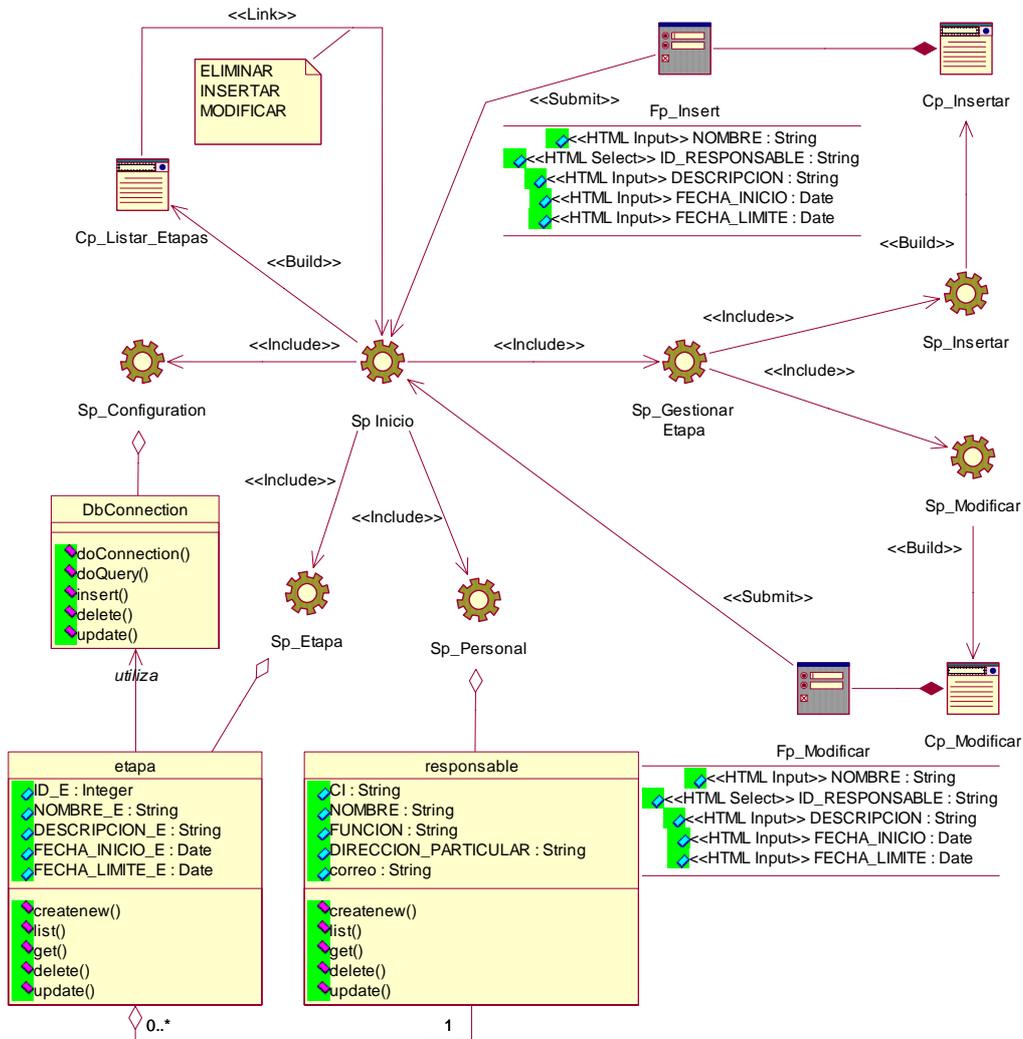


Figura 3.9 Caso de uso: Gestionar Etapa.

Gestionar Fase

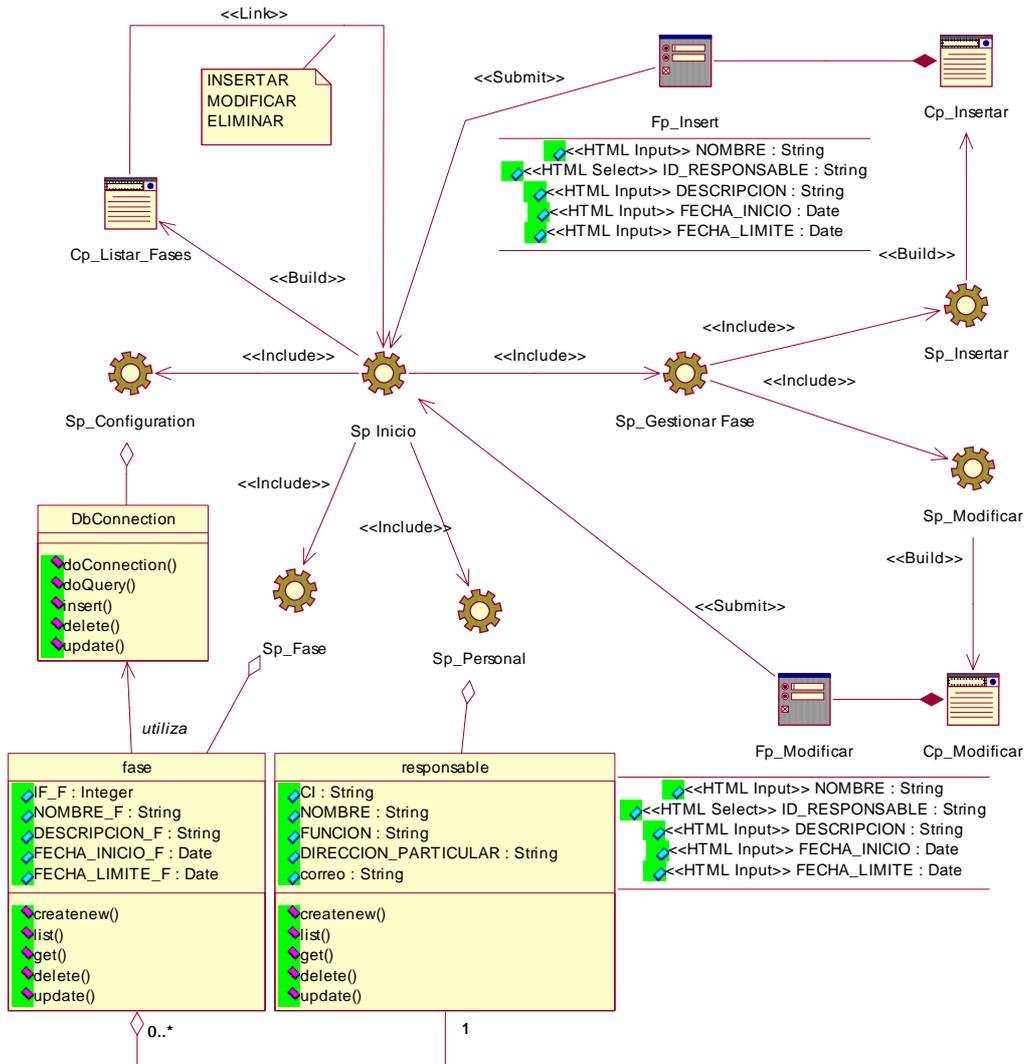


Figura 3.10 Caso de uso: Gestionar Fase.

Gestionar Personal

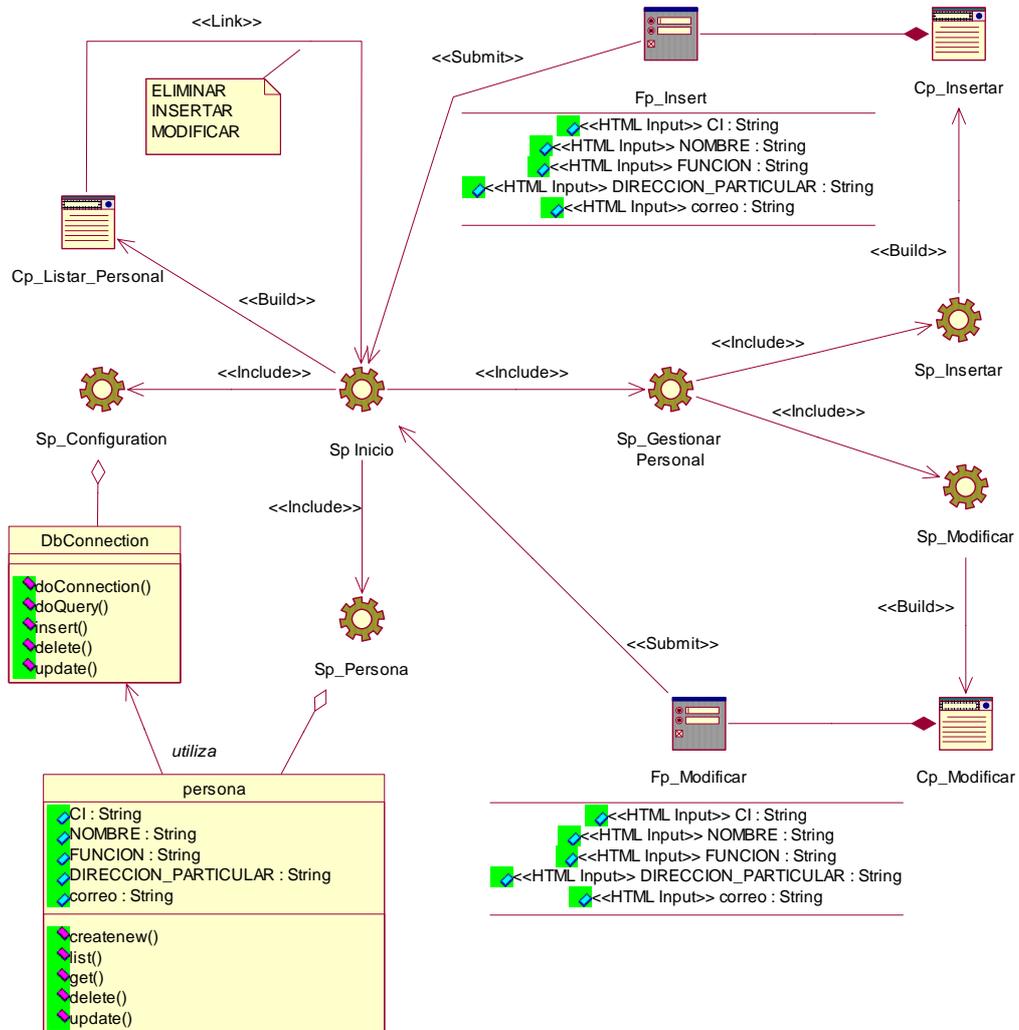


Figura 3.11 Caso de uso: Gestionar Personal.

Gestionar Flujo

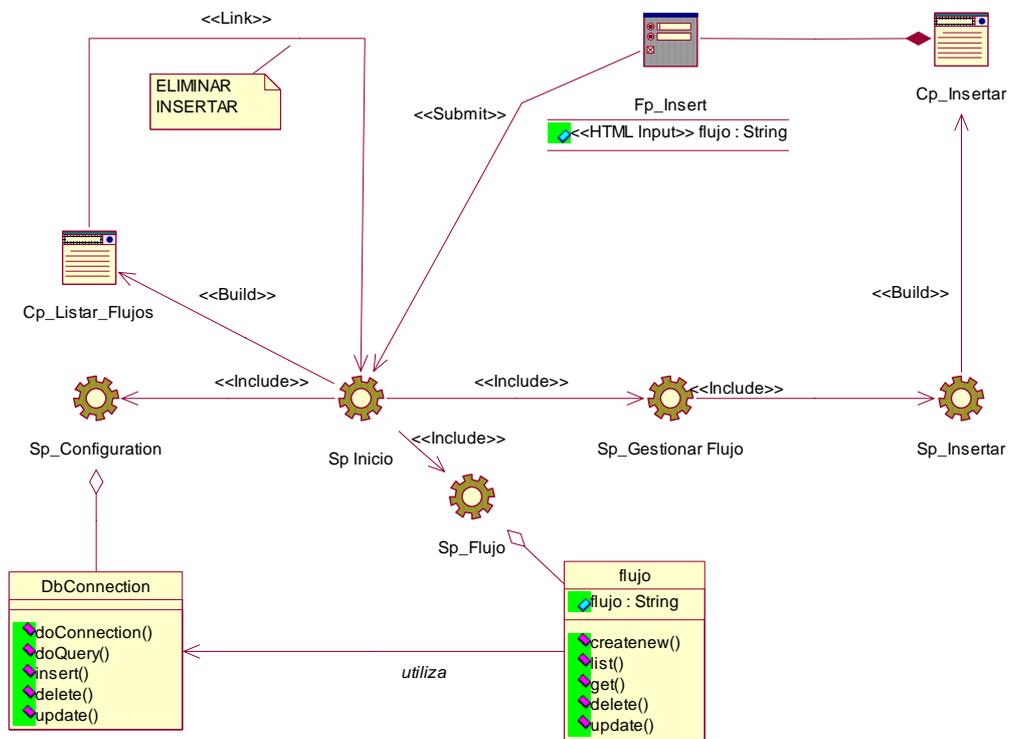


Figura 3.12 Caso de uso: Gestionar Flujo.

Gestionar Permiso

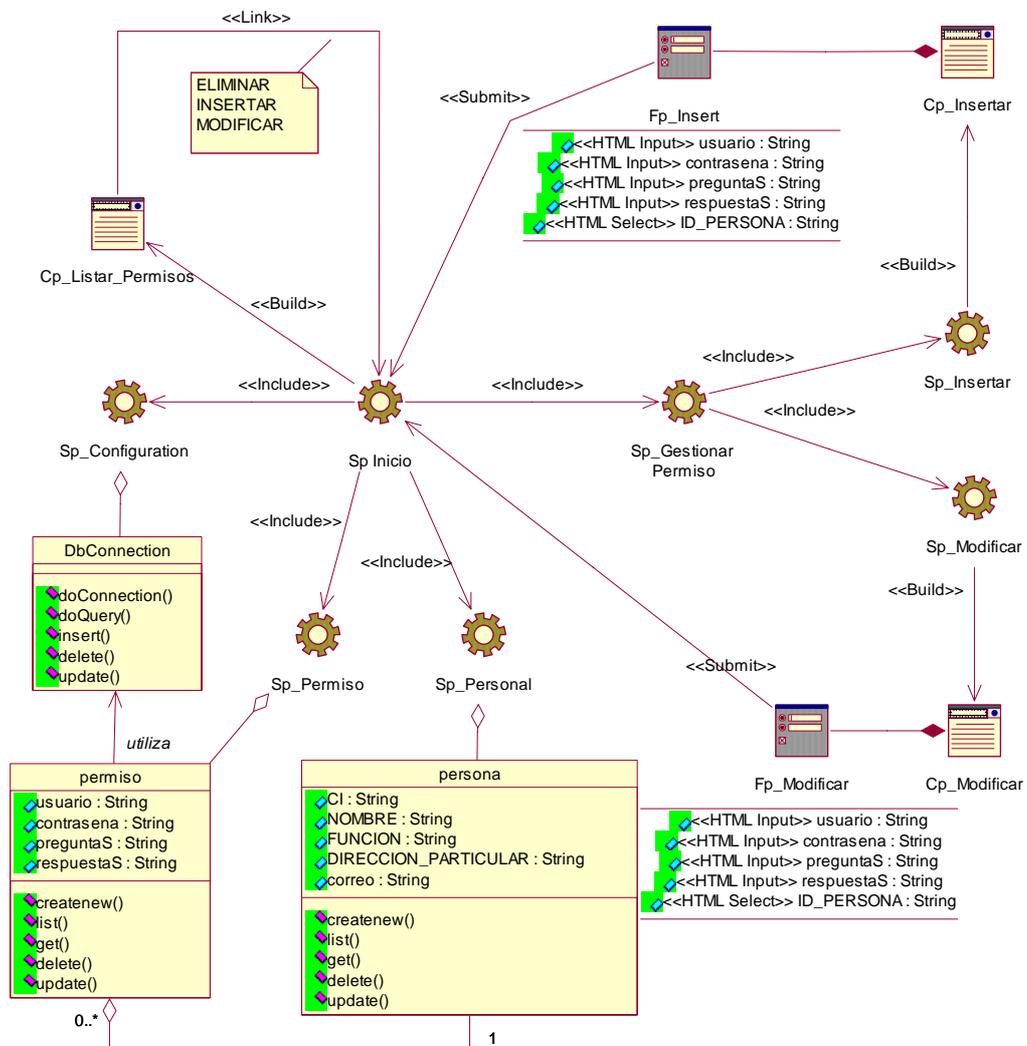


Figura 3.13 Caso de uso: Gestionar Permiso.

Generar Reporte Exportable

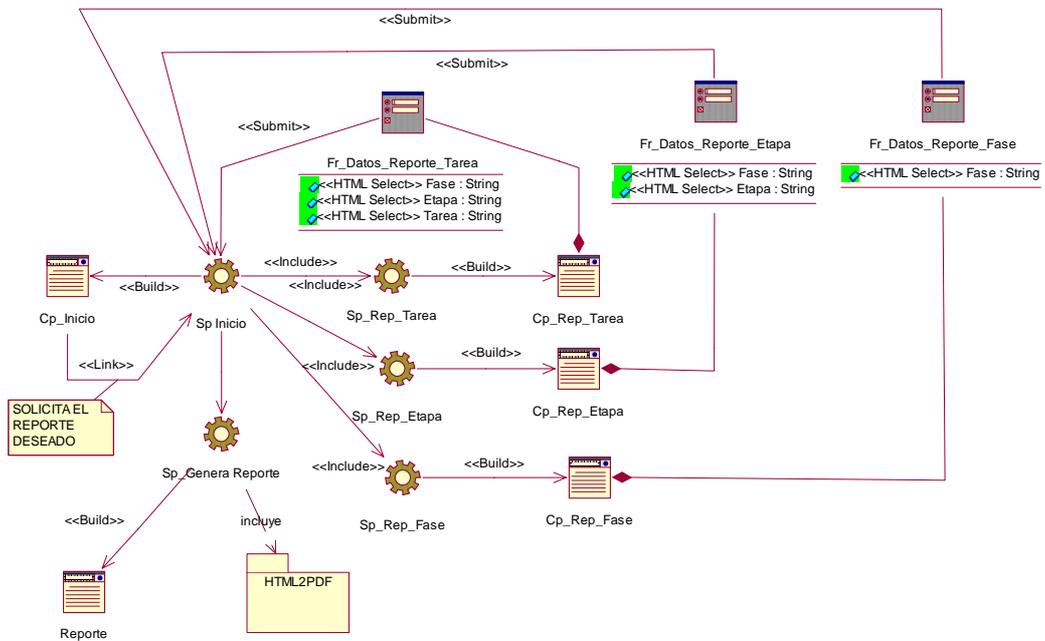


Figura 3.14 Caso de uso: Generar Reporte Exportable.

Visualizar Responsabilidades

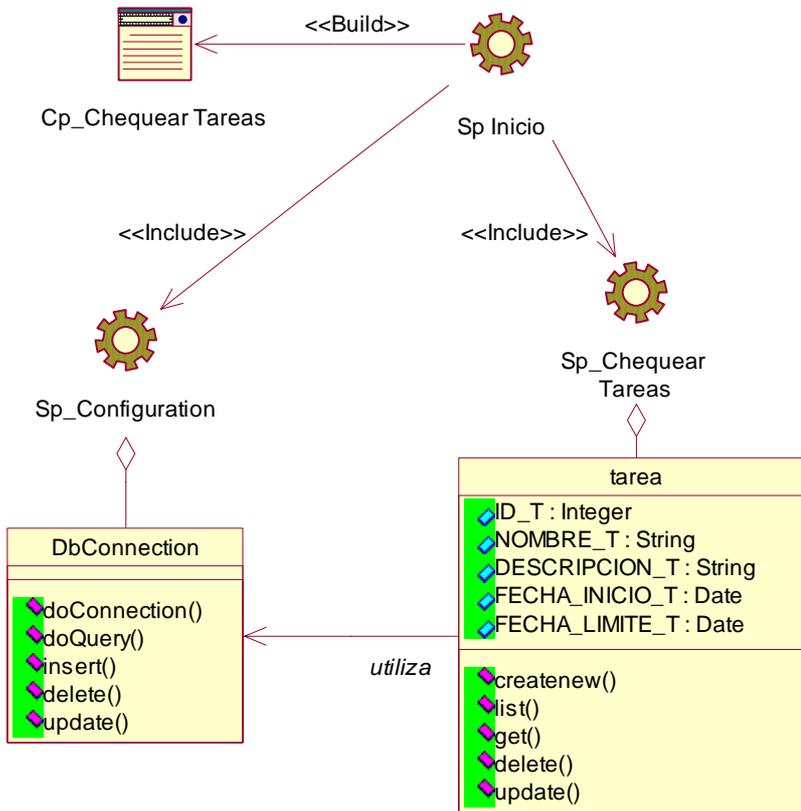


Figura 3.15 Caso de uso: Visualizar Responsabilidades.

Modificar Indicador

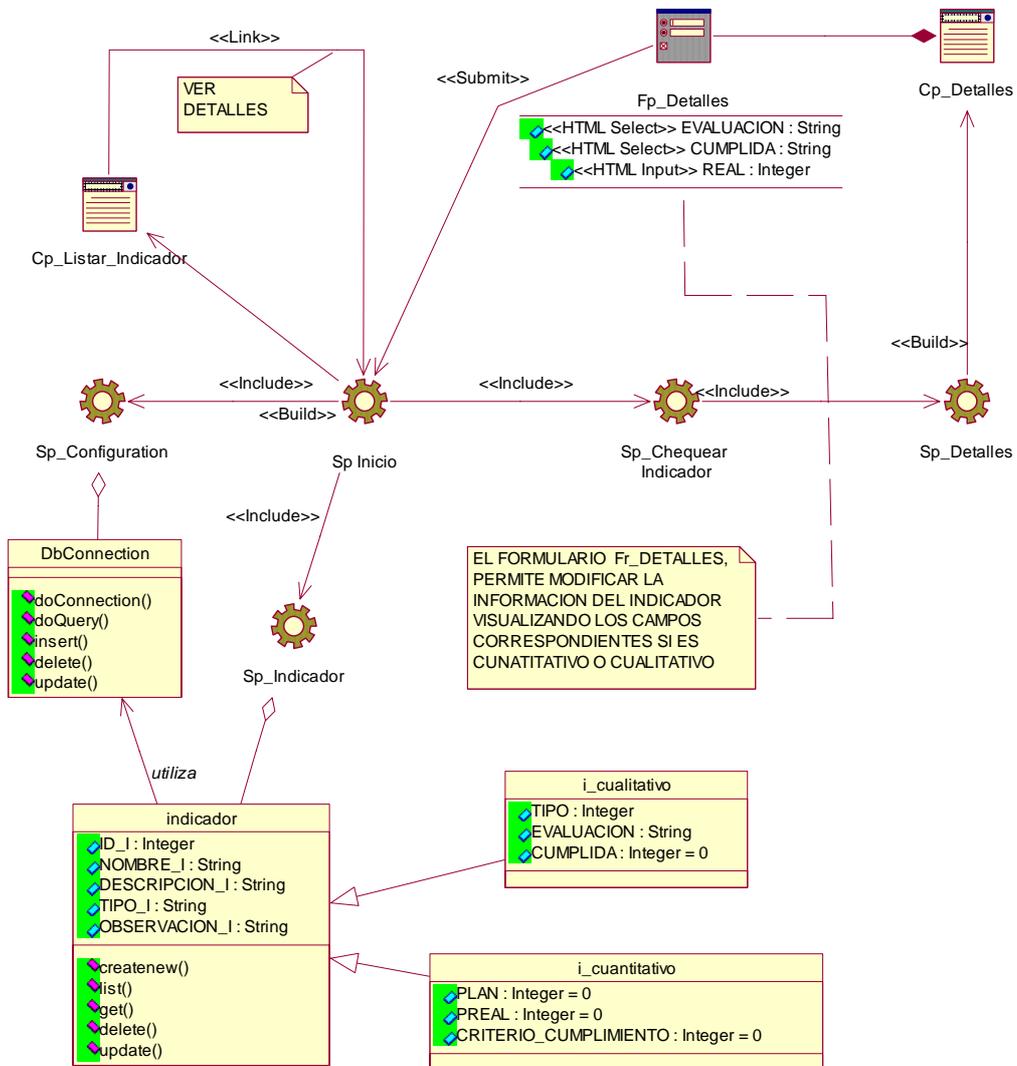


Figura 3.16 Caso de uso: Modificar Indicador.

Realizar Salva de Seguridad al Sistema de Gestión

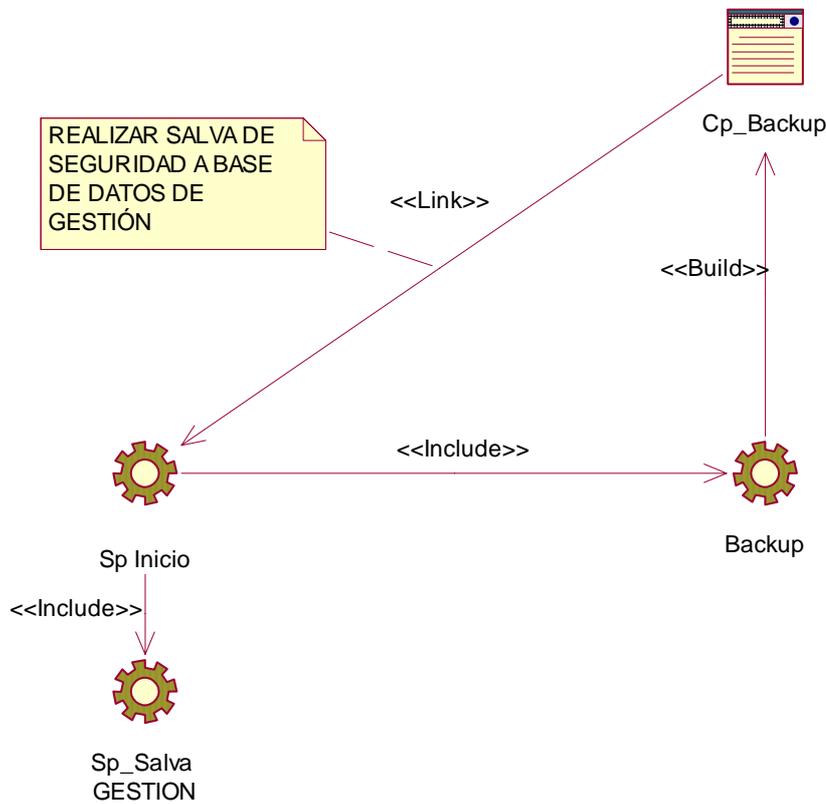


Figura 3.17 Caso de uso: Realizar Salva de Seguridad al Sistema de Gestión

Consultar la Ayuda

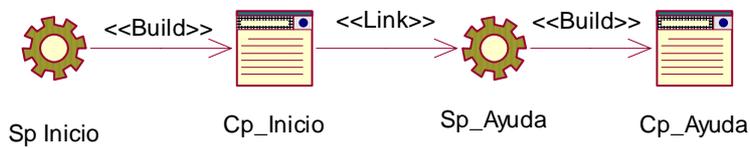


Figura 3.18 Caso de uso: Consultar la Ayuda.

Diagnosticar Equipo y Descargar Utilidad de Diagnóstico de Equipo

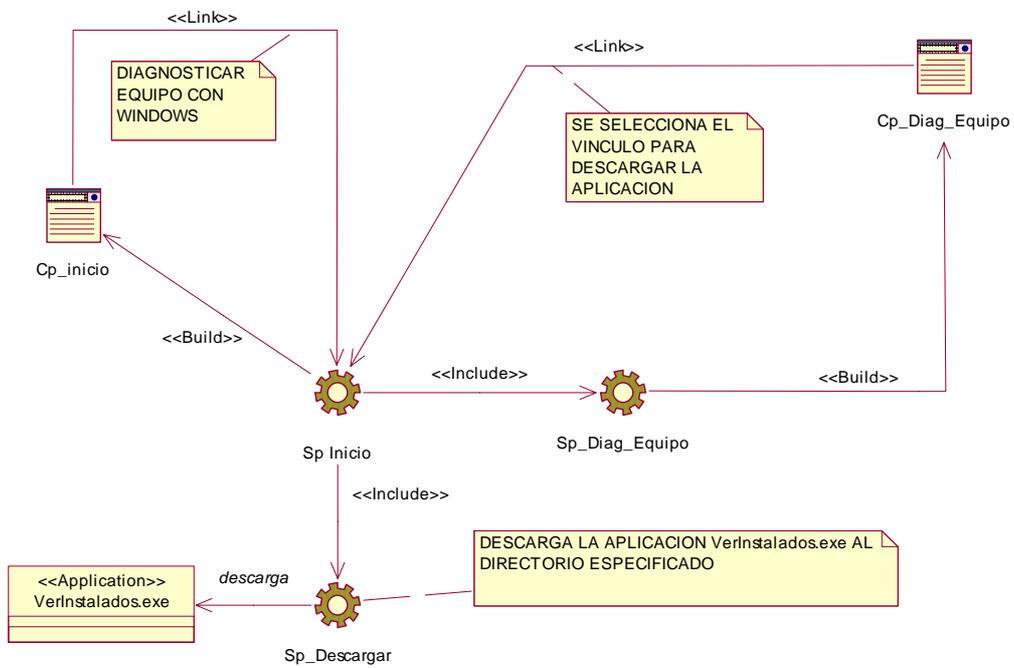


Figura 3.19 Casos de uso: Diagnosticar Equipo y Descargar Utilidad de Diagnóstico de Equipo.

Subir Diagnóstico Realizado a Equipo

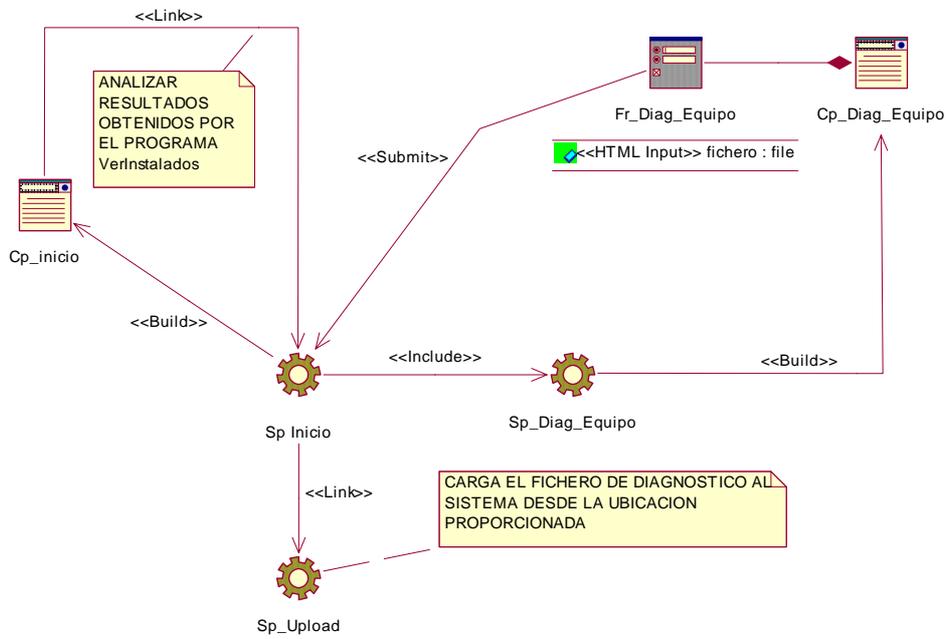


Figura 3.20 Caso de uso: Subir Diagnóstico Realizado a Equipo.

Eliminar Diagnóstico Inecesario

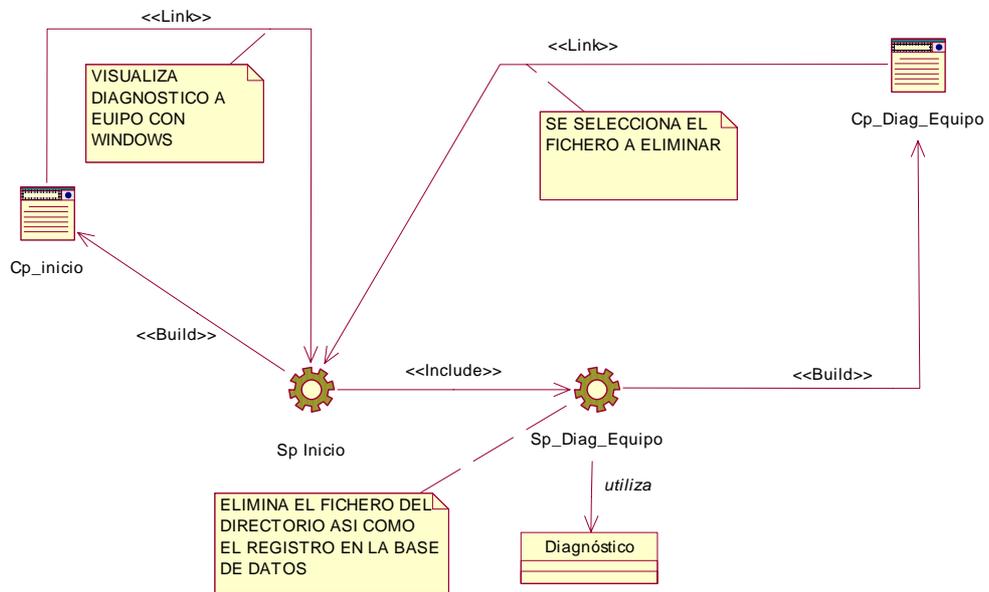


Figura 3.21 Caso de uso: Eliminar Diagnóstico Inecesario.

Visualizar Diagnóstico Realizado a Equipo

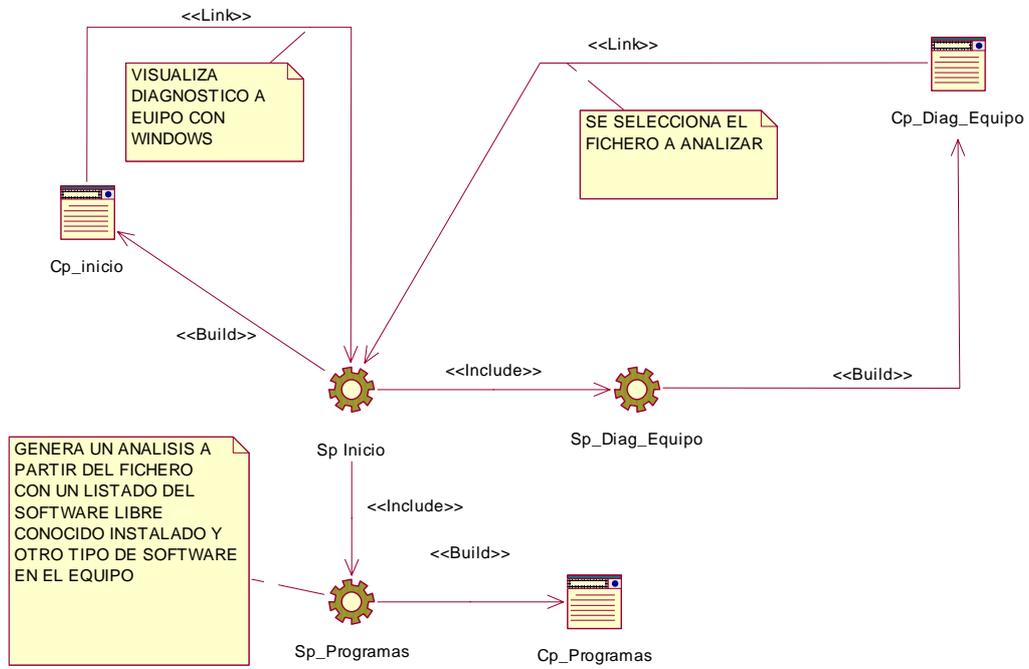


Figura 3.22 Caso de uso: Visualizar Diagnóstico realizado a Equipo

Diagnosticar Red

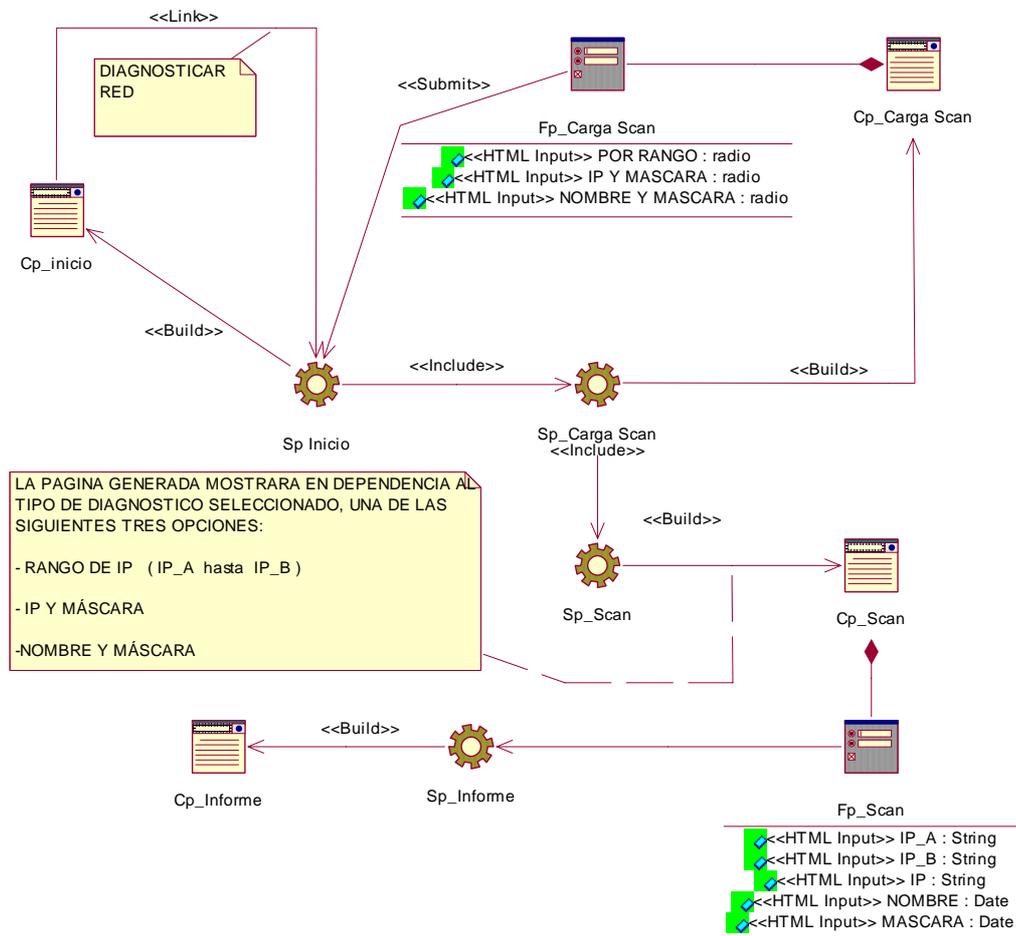


Figura 3.23 Caso de uso: Diagnosticar Red.

Realizar Salva de Seguridad al Registro de Diagnósticos

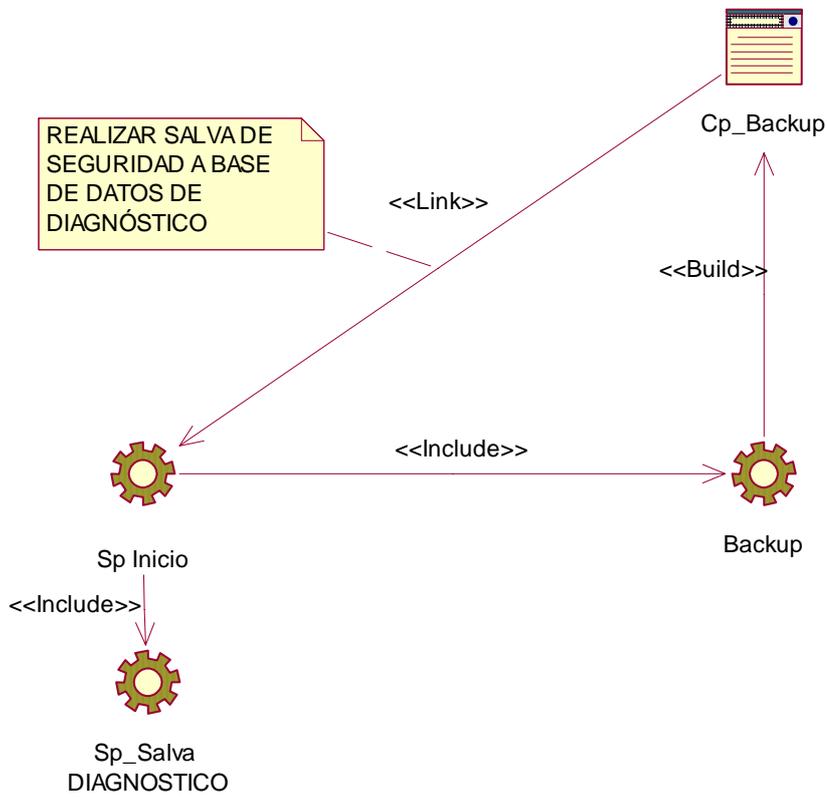


Figura 3.24 Realizar Salva de Seguridad al Registro de Diagnósticos.

3.3 Principios de diseño.

De manera general el diseño está orientado a facilitar la rapidez y eficiencia en su utilización. En todo momento se asegura funcionalidad y comodidad para las acciones correspondientes a la manipulación de la información. El diseño de la interfaz, el formato de los reportes y el tratamiento de las excepciones son determinantes en el éxito o fracaso de una aplicación Web.

3.3.1 Estándares en la interfaz de la aplicación.

En todo el sistema se contrastan colores con tonalidad azul oscuro, azul claro, blanco y naranja. El color azul en contraste con el blanco favorece un ambiente fresco, limpio y deseable. El naranja en los grupos de menú implica acción y funcionalidad.

La letra debe ser legible y de un color que le brinde perceptibilidad en contraste con el fondo. El uso de imágenes se limita a las necesarias evitando las animaciones y efectos visuales que además afectan la rapidez de la aplicación.

De esta manera se garantiza un diseño único en todo el sistema con características que amenizan la interacción con el mismo.

Por su parte, los reportes se construyen teniendo en cuenta que pueden ser reproducidos en impresoras de baja calidad. Se dispone un título adecuado en la parte superior de cada informe incorporando, además, la fecha en que se genera, lo cuál resulta de utilidad dadas las características de la aplicación.

Se ubica en la parte derecha el menú principal con las opciones correspondientes a cada usuario. En todos los casos las funciones fundamentales como ir de regreso a la página de bienvenida, cambiar la contraseña personal, consultar la

ayuda y salir del sistema; se presentan en la parte superior del menú visualizado garantizando así la inmediata ubicación y familiarización con la aplicación.

El sistema permite la visualización de la información almacenada con el objetivo de facilitar las operaciones a realizar con la misma, para ello se mantiene el diseño antes mencionado y se adicionan las opciones necesarias en cada caso. La información obtenida a partir de la base de datos se organiza mediante la utilización de tablas.

3.3.2 Concepción general de la ayuda.

El sistema cuenta con una ayuda general la cual se encuentra disponible en cualquier momento. Es desarrollada siguiendo los mismos principios del diseño utilizados en la aplicación en cuestión aunque se muestra en una página independiente.

Está estructurada de manera tal que permite una inmediata orientación y fácil navegación por cada uno de sus tópicos. De esta forma se hace fácil consultar cualquier duda sobre el sistema.

3.3.3 Tratamiento de errores.

Se evita continuamente la introducción de errores. Con este propósito se muestra a cada usuario la información asociada a su responsabilidad. Las opciones de selección de listas se utilizan continuamente para minimizar la entrada de errores por teclado. Además presenta mecanismos para validar la información evitando la introducción de información errónea. Cuando se presenta algún error el mismo es comunicado con un lenguaje entendible mediante ventanas de alerta o dinámicamente incrustados en la misma aplicación.

3.4 Diseño de la base de datos.

La base de datos se diseña teniendo en mente garantizar que los datos persistentes son almacenados consistentemente. Con este fin se desarrollan los modelos lógico y físico de la base de datos.

3.4.1 Modelo lógico de datos.

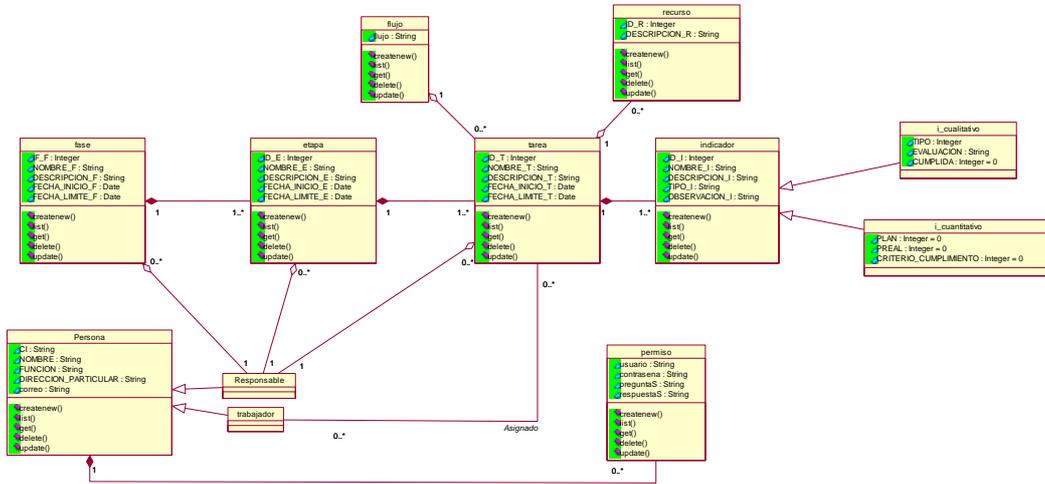


Figura 3.25 Modelo Lógico de Datos.

3.4.2 Modelo físico de datos.

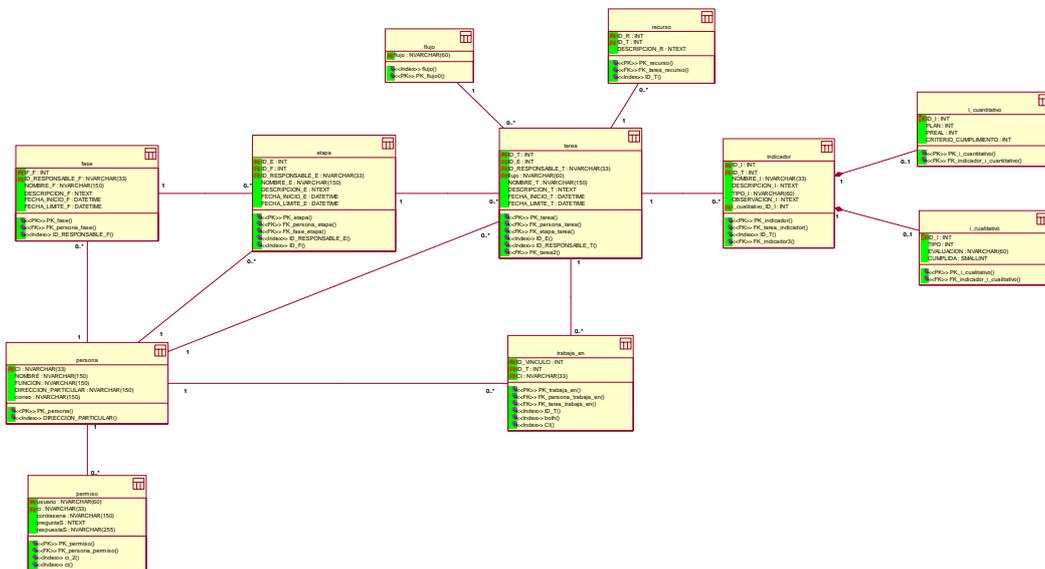


Figura 3.26 Modelo Físico de Datos.

3.5 Diagrama de implementación.

Este diagrama representa cómo se implementa el sistema en términos de componentes y subsistemas de implementación. Se describe la manera en que son organizados los componentes en el entorno de la implementación y se muestra la dependencia que existe entre los componentes.

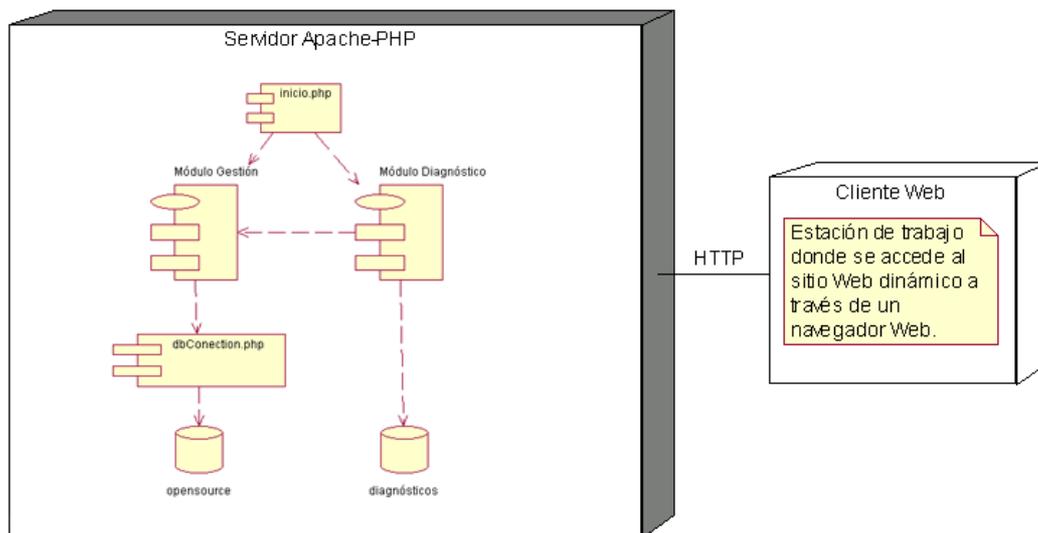


Figura 3.27 Diagrama de Implementación.

3.7 Estudio de factibilidad

La planificación es una actividad de gran importancia en el desarrollo de software. Para que la misma se logre efectuar exitosamente resulta fundamental evaluar el sistema de software por desarrollar en cuanto a su nivel de dificultad buscando obtener un aproximado del tiempo y el costo necesarios para la elaboración del mismo. Se decidió efectuar la estimación por puntos de caso de uso. A continuación se realiza el estudio que concluye si es o no factible la realización del proyecto, basado en un previo análisis de costos y beneficios.

Estimación por puntos de casos de uso

Cálculo de puntos de casos de uso sin ajustar.

$$UUCP = UAW + UUCW$$

UUCP: Puntos de casos de uso sin ajustar.

UAW: Factor de peso de los actores sin ajustar.

UUCW: Factor de peso de los casos de uso sin ajustar.

Para obtener el factor de peso de los actores si ajustar (UAW) se asigna un valor a cada tipo de actor

Tipo	Descripción	Factor	Actores
Simple	Otro sistema que interactúa con el sistema a desarrollar mediante una interfaz de programación (API).	1	
Medio	Otro sistema interactuando a través de un protocolo (ej. TCP/IP) o una persona interactuando a través de una interfaz en modo texto.	2	
Complejo	Una persona que interactúa con el sistema mediante una interfaz gráfica (GUI).	3	4

Tabla 3.1 Factor de Peso de los Actores.

La fórmula es: $UAW = \text{Sum}(\text{cantidadDeUnTipoDeActor} * \text{Factor})$

$$UAW = 12$$

Para obtener el factor de peso de los casos de uso sin ajustar (UUCW) se analiza la complejidad de cada caso de uso. La complejidad se establece teniendo en cuenta la cantidad de transacciones que se efectúan. Donde una transacción es una secuencia de actividades atómicas, es decir que o se efectúa la secuencia completa o no se efectúa. En la siguiente tabla se dividen los casos de uso el sistema de acuerdo a su nivel de complejidad.

Tipo de caso de uso	Descripción	Factor	Casos de Uso
Simple	3 transacciones o menos	5	16
Medio	4 a 7 transacciones	10	9
Complejo	Más de 7 transacciones	15	-

Tabla 3.2 Factor de Peso de los Casos de Uso.

La fórmula es: $UUCW = \text{Sum}(\text{CantidadDeUnTipoDeCasoUso} * \text{Factor})$

$$UUCW = 16 \cdot 5 + 9 \cdot 10$$

$$UUCW = 170$$

$$UUCP = UAW + UUCW$$

$$UUCP = 12 + 170$$

$$UUCP = 182$$

Cálculo de puntos de casos de uso ajustados.

El valor UUCP se debe ajustar mediante:

$$UCP = UUCP \cdot TCF \cdot EF$$

UCP: Puntos de casos de uso ajustados.

TCF: Factor de complejidad técnica.

EF: Factor de ambiente.

El TCF (factor de complejidad técnica) se calcula mediante la cuantificación de un conjunto de factores que determinan la complejidad técnica del sistema. Cada factor se cuantifica en un valor desde 0 (aporte irrelevante) hasta 5 (aporte muy relevante)

Factor	Descripción	Peso	Valor Asignado	Total
T1	Sistema distribuido.	2	0	0
T2	Objetivos de performance o tiempo de respuesta.	1	4	4
T3	Eficiencia del usuario final.	1	1	1
T4	Procesamiento interno complejo.	1	2	2
T5	El código debe ser reutilizable.	1	4	4
T6	Facilidad de instalación.	0.5	4	2
T7	Facilidad de uso.	0.5	3	1.5
T8	Portabilidad.	2	4	8
T9	Facilidad de cambio.	1	3	3
T10	Concurrencia.	1	0	0
T11	Incluye objetivos especiales de seguridad.	1	3	3
T12	Provee acceso directo a terceras partes.	1	5	5
T13	Se requiere facilidades especiales de entrenamiento a usuario.	1	1	1

Tabla 3.3 Factor de Complejidad Técnica.

Las fórmulas para este punto son:

$$TFactor = \text{Sum}(\text{Valor} * \text{Peso})$$

$$TFactor = 34.5$$

$$TCF = 0.6 + (0.01 * TFactor)$$

$$TCF = 0.95$$

El factor de ambiente (EF) está relacionado con las habilidades y entrenamiento del grupo de desarrollo que realiza el sistema. Cada factor se cuantifica con un valor desde 0 (aporte irrelevante) hasta 5 (aporte muy relevante).

Factor	Descripción	Peso	Valor	Total
E1	Familiaridad con el modelo de proyecto utilizado.	1.5	4	6
E2	Experiencia en la aplicación.	0.5	3	1.5
E3	Experiencia en orientación a objetos.	1	4	4
E4	Capacidad del analista líder.	0.5	4	2
E5	Motivación	1	4	4
E6	Estabilidad de los requerimientos	2	2	4
E7	Personal part-time	-1	2	-2
E8	Dificultad del lenguaje de programación	-1	2	-2

Tabla 3.4 Factor de Ambiente.

Las fórmulas para este punto son:

$$EFactor = \text{Sum}(\text{Valor} * \text{Peso})$$

$$EFactor = 17.5$$

$$EF = 1.4 + (-0.03 * EFactor)$$

$$EF = 0.88$$

$$UCP = UUCP * TCF * EF$$

$$UCP = 182 * 0.95 * 0.88$$

$$UCP = 152.15$$

Estimación de esfuerzo a través de los puntos de casos de uso.

$$E = UCP + CF$$

E: Esfuerzo estimado en horas hombres.

CF: Factor de conversión

Este cálculo se realiza con el fin de tener una aproximación del esfuerzo, pensando solo en el desarrollo según las funcionalidades de los casos de uso. Anteriormente, se sugería utilizar 20 horas persona por UCP, pero a través del tiempo se ha ido mejorando. Está basado en los factores ambientales y se calcula de la siguiente manera:

Primero se debe contar la cantidad de factores ambientales del E1 al E6 que tienen una puntuación menor a 3, también contar la cantidad de estos mismos del E7 y E8 que son mayores que 3.

Factor	Filtro
De E1 a E6	Factor < 3
De E7 a E8	Factor > 3

- Para evaluar el resultado o la cantidad total según la siguiente tabla:

Horas-Persona (CF)	Descripción
20	Si el valor es <=2
28	Si el valor es <=4
36	Si el valor es >=5

Tabla 3.5 Horas – Persona.

$$CF = 20 \text{ Horas-Hombre} / \text{Punto de Casos de uso.}$$

Esfuerzo:

$$E = UCP * CF$$

$$E = 152.15 * 20$$

$$E = 3043$$

El resultado (E) constituye el esfuerzo estimado en la programación del proyecto y representa el 40 % del esfuerzo total.

$$ET = E / 0.4$$

ET: Esfuerzo total estimado para el desarrollo del proyecto.

$$ET = 3043 / 0.4$$

$$ET = 7607.5$$

Actividad	Porcentaje	Horas - Hombre
Análisis	10%	760.75
Diseño	20%	1521.5
Programación	40%	3043
Pruebas	15%	1141.13
Sobrecarga	15%	1141.13
Total	100%	7607.5

Tabla 3.6 Distribución del Esfuerzo.

Tiempo de desarrollo

$$TDes = ET / CH$$

TDes: Tiempo de desarrollo.

CH: Cantidad de hombres. Se cuenta con una persona para la realización del proyecto.

$$TDes = 7607.5 / 2$$

$$TDes = 3803.75 \text{ Horas}$$

Costo del proyecto

$$CT = ET * CH * TH$$

CT: Costo Total del proyecto.

TH: Tarifa horaria asumiendo el salario básico mensual de \$ 225 (1.171).

$$CT = 3803.75 * 2 * 1.171$$

$$CT = \$ 8908.38$$

Beneficios Tangibles e Intangibles

El sistema propuesto permite una adecuada gestión de la información en el proceso de migración a software libre. Implica un ahorro substancial en tiempo y escritorio favoreciendo la coordinación y el control del progreso del proceso. Permite efectuar diagnósticos en tiempo real a la red de computadoras y conocer el estado actual del software instalado sin necesidad de presencia física en las instalaciones correspondientes. Ofrece reportes digitales que pueden ser impresos

pero, además, ser almacenados en este soporte, lo que repercute en ahorro de recursos. Logra integridad y seguridad en la información almacenada.

Análisis de costos y beneficios

Para justificar el desarrollo de un producto informático es necesario analizar los beneficios que traería consigo su implementación así como los costos. Por esta razón se consideran los siguientes aspectos:

- El costo de desarrollo asciende a \$ 8908.38 con la participación de 2 personas en un tiempo aproximado de 11 meses.
- No se necesita adquirir nuevos equipos computacionales para la implantación del sistema ya que la entidad cuenta con el equipamiento necesario.
- El sistema está basado en un funcionamiento sobre tecnologías de software libre por lo que para su instalación, mejoras y futuras versiones no será necesario el pago de licencias para software asociados.
- El sistema ayudará a mejorar el trabajo en el proceso de migración a software libre en la Universidad de Cienfuegos.

Teniendo en cuenta lo anteriormente planteado podemos decir que los beneficios al implantar el sistema son mayores que los costos.

3.7 Conclusiones.

En este capítulo fueron presentados los diagramas de clases Web, los principios del diseño seguidos en el sistema propuesto, la concepción general de la ayuda y el tratamiento de excepciones. También se expuso el diseño de la base de datos a partir del modelo lógico y físico de datos. A través del diagrama de implementación quedan mostrados los elementos fundamentales de la implementación del sistema. Además, se realizó el estudio de factibilidad correspondiente al desarrollo del proyecto. Utilizando el método de estimación por puntos de casos de uso se realizó el análisis entre los costos y los beneficios que reporta la aplicación concluyendo que su realización es económicamente factible.

Conclusiones.

La gestión de la información en un proceso de Migración a Software Libre trae consigo una serie de operaciones que de ser realizadas manualmente resultarían en una engorrosa y lenta tarea. Dada esta situación, podemos afirmar que la realización de un sistema informático basado en las necesidades existentes agilizaría de forma notable el manejo de los datos concernientes al proceso de migración.

Luego de un estudio de las características de las estrategias de migración existentes se presenta una propuesta de sistema que basa su funcionamiento en los estándares de la W3C para la Web. Esta propuesta permite a los usuarios el acceso mediante la red a los módulos de gestión y diagnóstico desarrollados, garantizando así una mejor coordinación y dinamismo en el trabajo.

La aplicación está provista de un ambiente cómodo, fácil de entender, que cumple los estándares de diseño para Web y utiliza técnicas de programación orientada a objetos. Se utilizó como servidor de aplicaciones Apache y PHP como lenguaje de programación del lado del servidor, esto permitió lograr un sistema multiplataforma capaz de funcionar sobre diferentes sistemas operativos.

Se cumplió con el objetivo general del trabajo siguiendo la metodología RUP, y utilizando el lenguaje UML para la modelación del sistema.

Recomendaciones

Aspectos importantes a tener en cuenta para un futuro enriquecimiento de este trabajo serían:

- Conservar la concepción de un sistema multiplataforma dado el propósito con que se desarrolla esta aplicación.
- Conservar la flexibilidad del sistema de modo que pueda ser aplicado a diversas instituciones independientemente de las características particulares de las mismas.
- Valorar la incorporación de nuevos métodos que puedan ir apareciendo en el aspecto de la detección de sistemas operativos.
- Ampliar el modulo de reportes del sistema con el fin de brindar mayores prestaciones a los usuarios.

Referencias Bibliográficas.

- Iñaki I. Rojo . (2009, Abril 2). Open Source: los programas íntegros. Recuperado Abril 2, 2009, a partir de <http://www.baquia.com/com/legacy/8512.html>.
- Aileen Morrison. (2009, Mayo). Servidor Web Apache. Conectiva Linux Chile. Recuperado a partir de <http://linux.ubiobio.cl/pasados/segundo/documentacion/ServidorWebApache.pdf.gz>.
- Alemania es el país con mayor uso de software libre . (2009). . Recuperado Abril 2, 2009, a partir de <http://www.softwarelibre.cl/drupal/?q=node/1059>.
- Andrew Orłowski in San Francisco. (2003, Diciembre). Israel accelerates free software migration . Recuperado a partir de http://www.theregister.co.uk/Israel_accelerates_free_software_migration_The_Register.htm.
- Arquitectura Web. (2009, Abril). . Recuperado a partir de www.di.uniovi.es/~dflanvin/docencia/dasdi/teoria/Transparencias/06_Arquitectura_Web.pdf.
- David Barroso Berrueta. (2003). Un enfoque práctico para engañar la detección remota de SO de Nmap. Recuperado Abril 2, 2009, a partir de http://his.sourceforge.net/proy_his/papers/nmap/nmap.es.html.
- Default TTL Values in TCP/IP. (1999, Abril 8). . Recuperado Abril 3, 2009, a partir de http://secfr.nerim.net/docs/fingerprint/en/ttl_default.html.
- Definición de Sistema Operativo. (2009, Mayo). . Recuperado Mayo 7, 2009, a partir de <http://www.masadelante.com/faq-sistema-operativo.htm>.
- El Gobierno de Brasil ha dado un paso más en favor de las tecnologías de código abierto. (2008, Mayo). . Recuperado a partir de <http://www.gnumax.net>.
- EMS SQL Manager for MySQL. (2008, Febrero). . Recuperado a partir de <http://www.freedownloadscenter.com/es/Negocio/Aplicaciones/>.
- Esteban Trigos García. (2009, Mayo). Guía Práctica para usuarios. PHP. Recuperado a partir de <http://dialnet.unirioja.es/servlet/libro?codigo=242312>.
- Fernando Gont. (2008, Noviembre 25). Resultados de un análisis de seguridad de los protocolos TCP e IP. Recuperado Abril 2, 2009, a partir de www.gont.com.ar/talks/CIEER08/fgont-cieer-resultados-analisis-de-seguridad-tcp-ip.ppt.

Referencias Bibliográficas.

- Fundación Código Libre Dominicano. (s.d.). . Recuperado Abril 2, 2009, a partir de <http://www.codigolibre.org/modules.php?name=Sections&op=viewarticle&artid=129&page=2>.
- Gerard Farras. (2009, Abril 2). Utilidades y Técnicas para la determinación de un host remoto. Recuperado Abril 3, 2009, a partir de <http://www.km103.com/gfarras/articles/OSDetection/#m33>.
- Ing. Yoandy Pérez Villazón, & Ing. Ramón Paumier Samón. (2009). Metodología para la Migración a Software Libre de las Universidades del MES. Informatica 2009.
- Internet Explorer - Wikipedia, la enciclopedia libre. (2009, Mayo). . Recuperado Mayo 6, 2009, a partir de http://es.wikipedia.org/wiki/Internet_explorer.
- Introducción a Nmap. (2003, Agosto 19). . Recuperado Abril 2, 2009, a partir de <http://www.maestrosdelweb.com/editorial/nmap/>.
- Iván Nieto Pérez. (2009, Mayo). Diccionario de siglas. Recuperado Mayo 7, 2009, a partir de <http://www.elcodigo.net/tutoriales/diccionario.html>.
- Ivar Jacobson, Grady Booch, & Jame Rumbaugh. (s.d.). *El Proceso Unificado de Desarrollo de Software*. Pearson Educacion, Nuñez de Balboa, 12028006, Madrid.
- JavaScript - Wikipedia, la enciclopedia libre. (2009, Mayo). . Recuperado Mayo 7, 2009, a partir de <http://es.wikipedia.org/wiki/Javascript>.
- Jose Aguilar. (s.d.). El Software Libre: un nuevo fenómeno informático. Recuperado Marzo 31, 2009, a partir de <http://www.taringa.net/posts/linux/2022363/Software-libre-ques-por-Jose-Aguilar.html>.
- José Luis Murillo. (2009, Marzo 23). Importancia del Software libre en la educación - edulibre.info. Recuperado Abril 2, 2009, a partir de <http://edulibre.info/spip.php?article130>.
- Lianet Arias Sosa. (2009, Febrero 10). Desarrollo y Soberanía. Recuperado Abril 2, 2009, a partir de <http://www.granma.cubaweb.cu/2009/02/10/nacional/artic02.html>.
- Marcos Pérez Ramírez. (2003, Agosto). CONOZCA EL SOFTWARE LIBRE. Recuperado Mayo 15, 2009, a partir de http://www.universia.pr/portada/actualidad/noticia_actualidad.jsp?noticia=11275.
- Migracion. (s.d.). . Recuperado a partir de <http://www.eiraworks.com/migracion-a-software-libre>.

Referencias Bibliográficas.

- Mozilla Firefox - Wikipedia, la enciclopedia libre. (2009, Mayo). . Recuperado Mayo 6, 2009, a partir de http://es.wikipedia.org/wiki/Mozilla_firefox.
- MySQL 5.0 Reference Manual. (2009, Mayo). . Recuperado Mayo 6, 2009, a partir de <http://dev.mysql.com/doc/refman/5.0/es/introduction.html>.
- Ofir Arkin, & Fyodor Yarochkin. (2002, Agosto). Xprobe v2.0 A “Fuzzy” Approach to Remote Active Operating System Fingerprinting.pdf.
- Oscar Muñoz. (2007). Arquitectura de aplicaciones Web. Conferencia de Seminarios Especiales I.--Cienfuegos: UCF .
- phpDesigner. (s.d.). . Recuperado Mayo 7, 2009, a partir de <http://php-designer.aplicaciones.movistar.es/>.
- PHPMyAdmin. (2008, Febrero). . Recuperado a partir de <http://www.desarrolloweb.com/articulos/844.php>.
- PROYECTOS. (s.d.). . Recuperado Mayo 7, 2009, a partir de <http://www.usmp.edu.pe/publicaciones/boletin/fia/info36/proyectos.html>.
- Real Academia Española. (2009, Mayo). . Recuperado a partir de www.rae.es.
- Software Libre en Venezuela. (2009, Abril 2). . Recuperado Abril 2, 2009, a partir de <http://www.softwarelibre.venezuela.net.ve/>.
- SQL - Wikipedia, la enciclopedia libre. (2009, Mayo). . Recuperado Mayo 7, 2009, a partir de <http://es.wikipedia.org/wiki/SQL>.
- Xavier Ferré Grau, & María Isabel Sánchez Segura. (2009, Marzo 31). Desarrollo Orientado a Objetos con UML. Recuperado Marzo 31, 2009, a partir de <http://www.clikear.com/manuales/uml/introduccion.aspx>.
- Yoandry Pacheco Aguila. (2007, Febrero). AJAX un nuevo acercamiento a las aplicaciones Web. Recuperado a partir de <http://www.monografias.com/trabajos43/ajax/ajax.zip>.
- Yolanda de J. Ofarril Dinza. (2009, Mayo). Sistema Entrenador Inteligente con Tecnología Multimedia. *Revista Latinoamericana de Investigación en Matemática Educativa, Volumen 3*. Recuperado Mayo 7, 2009, a partir de <http://redalyc.uaemex.mx/redalyc/pdf/335/33530202.pdf>.

Referencias Bibliográficas.

Yudivián Almeida Cruz. (2008, Abril 12). Ecuador decreta migrar a Software Libre. Recuperado a partir de <http://profesores.matcom.uh.cu/%7Eyudy/blog/archives/18-Ecuador-decreta-migrar-a-Software-Libre.html>.

Bibliografía.

Arquitectura de Aplicaciones para la Web.pdf. (s.d.). . Recuperado Mayo 15, 2009, a partir de <http:// triana.escet.urjc.es/apliweb/AW-Tema4.pdf>.

Arquitectura_y_diseño_de_sistemas_web_modernos.pdf (application/pdf Objeto). (s.d.). . Recuperado Abril 3, 2009, a partir de http://www.cii-murcia.es/informas/ene05/articulos/Arquitectura_y_diseno_de_sistemas_web_modernos.pdf.

Categorías de Software Libre y no Libre - Proyecto GNU - Fundación para el Software libre (FSF). (s.d.). . Recuperado Febrero 14, 2009, a partir de <http://www.gnu.org/philosophy/categories.es.html>.

Diez ventajas del software libre y propietario :: ABADIA DIGITAL :: Noticias, Tecnología e Internet. (s.d.). . Recuperado Febrero 11, 2009, a partir de <http://www.abadiadigital.com/articulo/diez-ventajas-del-software-libre-y-propietario/>.

Ernesto De Spirito. (2000). Delphi: Accediendo al Registro de Windows. Recuperado Enero 16, 2009, a partir de <http://www.latumsoftware.com/es/delphi/00004.php>.

Filosofía del Proyecto GNU - Proyecto GNU - Free Software Foundation (FSF). (s.d.). . Recuperado Febrero 8, 2009, a partir de <http://www.gnu.org/philosophy/philosophy.es.html>.

Jim Conallen. (1999a). Building Web Applications with UML, Object Technology Series by Addison Wesley Longman.

Jim Conallen. (1999b, Junio). Modeling Web Application Architectures with UML, Rational Software. Recuperado a partir de http://www.rational.com/media/uml/resources/documentation/27662_webapps.pdf.

La Definición de Software Libre - Proyecto GNU - Free Software Foundation (FSF). (s.d.). . Recuperado Febrero 9, 2009, a partir de <http://www.gnu.org/philosophy/free-sw.es.html>.

- Nmap - Free Security Scanner For Network Exploration & Security Audits. (s.d.). .
Recuperado Abril 1, 2009, a partir de <http://nmap.org/>.
- Nmap Reference Guide. (s.d.). . Recuperado Noviembre 19, 2008, a partir de
<http://nmap.org/book/man.html>.
- OCS Inventory | Aprendiz de Linux. (s.d.). . Recuperado Febrero 2, 2009, a partir
de <http://aprendizdelinux.com/tag/ocs-inventory/>.
- OCS Inventory NG - Welcome to OCS Inventory NG web site ! (s.d.). . Recuperado
Febrero 7, 2009, a partir de <http://www.ocsinventory-ng.org/>.
- Open Source Initiative. (2007, Marzo). . Recuperado Enero 23, 2009, a partir de
<http://www.opensource.org/>.
- Rational Unified Process es una infraestructura flexible de desarrollo de software
que proporciona prácticas recomendadas probadas y una arquitectura
configurable. (s.d.). . Recuperado Noviembre 12, 2007, a partir de
<http://www.rational.com.ar/herramientas/rup.html>.
- Registro de Windows « Info Delphi. (s.d.). . Recuperado Marzo 19, 2009, a partir
de <http://infodelphi.wordpress.com/2009/01/29/registro-de-windows/>.
- Stan Ward, P. K. (s.d.). Building Web Solutions with the Rational Unified Process:
Unifying the Creative Design Process and the Software Engineering Process.
- UML: Diagramas UML. ¿Qué es UML?. Análisis y Diseño. Ingeniería del Software.
(s.d.). . Recuperado Mayo 9, 2008, a partir de
<http://www.ingenierosoftware.com/analisisydiseño/uml.php>.
- Unified Modeling Language. (s.d.). . Recuperado Diciembre 1, 2008, a partir de
<http://www.omg.org/technology/documents/formal/uml.htm>.
- xprobe - What is xprobe. (s.d.). . Recuperado Abril 11, 2009, a partir de
<http://linux.about.com/cs/linux101/g/xprobe.htm>.

Glosario de términos.

Ajax. Acrónimo de *Asynchronous JavaScript And XML*.

CSS. Cascading Style Sheets (Hojas de Estilo en Cascada).

HTML. HyperText Markup Language (Lenguaje de Marcado de Hipertexto).

Migración. Se refiere al proceso donde en una entidad determinada se rempazan los programas no libres por otros libres.

Open Source. (Código Abierto) Aplicaciones cuyo código fuente puede ser accedido libremente. Es una característica del software libre ser Open Source.

PHP. Hypertext Preprocessor (Preprocesador de Hipertexto).

Sistema Operativo. El sistema operativo es el software más importante de un ordenador. Es necesario para que funcionen los otros programas. Los sistemas operativos realizan tareas básicas, tales como enviar la información a la pantalla, tratamiento de archivos y directorios en el disco, así como el control de dispositivos periféricos (“Definición de Sistema Operativo,” 2009).

SMTP Simple Mail Transfer Protocol (Protocolo Simple para la Transferencia de Correos)

Software. Es un programa, o sea, un conjunto de instrucciones y reglas informáticas para ejecutar ciertas tareas en una computadora (“Real Academia Española,” 2009).

Software de Aplicación. Se refiere a aquel software que se ejecuta a través del Sistema Operativo y que está diseñado para facilitar al usuario la realización de una determinada labor.

Software Libre. Es aquel que una vez obtenido, puede ser usado, copiado, estudiado, modificado y redistribuido libremente.

SQL: Structured Query Language (Lenguaje de Consulta Estructurado).

Terminal (Ordenador). Se refiere a una computadora, o sea, una máquina electrónica la cual recibe y procesa datos para convertirlos en información útil y que para esto se vale de una gran variedad de secuencias o rutinas de instrucciones llamadas programas.

UML: Unified Modeling Language (Lenguaje Unificado de Modelado).

Anexos.

ANEXO 1 Algunos valores de TTL y el correspondiente Sistema Operativo (“Default TTL Values in TCP/IP,” 1999).

Version SO	Tcp_ttl	Udp_ttl
AIX	60	30
DEC Pathworks V5	30	30
FreeBSD 2.1R	64	64
HP/UX 9.0x	30	30
HP/UX 10.01	64	64
Iris 5.3	60	60
Iris 6.x	60	60
Linux	64	64
MacOS/MacTCP 2.0.x	60	60
OS/2 TCP/IP 3.0	64	64
OSF/1 V3.2 ^a	60	30
Solaris 2.x	255	255
SunOS 4.1.3/4.1.4	60	60
Ultrix V4.1/V4.2 ^a	60	30
VMS/Multinet	64	64
VMS/TCPware	60	64
VMS/Wollongong 1.1.1.1	128	30
VMS/UCX (latest rel.)	128	128
MS WfW	32	32
MS Windows 95	32	32
MS Windows NT 3.51	32	32
MS Windows NT 4.0	128	128

ANEXO 2 Prototipo De Caso de Uso Autenticar.

Este prototipo de pantalla de autenticación tiene un encabezado azul con el título "AUTENTICACIÓN". Contiene dos campos de texto: "Nombre de usuario:" y "Contraseña:". Debajo de los campos hay un botón azul que dice "ENTRAR" y un enlace de texto que dice "Recordar Contraseña".

ANEXO 3 Prototipo De Caso de Uso Recuperar Contraseña.

Este prototipo de pantalla para recuperar la contraseña tiene un encabezado azul con el título "AUTENTICACIÓN". Muestra la pregunta de seguridad: "Pregunta: el segundo nombre de tu papa". Hay un campo de texto con la respuesta "Manuel" y un botón gris que dice "Cambiar". En la parte inferior derecha hay un enlace de texto que dice "Regresar".

ANEXO 4 Prototipo De Caso de Uso Cambiar Contraseña.

Este prototipo de pantalla para cambiar la contraseña muestra el mensaje: "Cambio de contraseña para el usuario: inf2004102 . Por favor, Introduzca su Contraseña Anterior y luego teclee dos veces su Nueva Contraseña." Los campos incluyen: "Contraseña Anterior:" (oculta con puntos), "Contraseña Nueva:" (oculta con puntos), "Repetir Contraseña Nueva:" (oculta con puntos), "Pregunta de Seguridad:" (con el texto "el segundo nombre de tu papa"), "Respuesta:" (con el texto "Manuel") y un botón gris "Cambiar Contraseña" al final.

ANEXO 5 Prototipo De Caso de Uso Gestionar Indicador.

Listar

Nombre del Indicador	Cumplimiento	Realizado	Insertar Indicador
Indicadores de Tipo Cuantitativo			
Progreso del Levantamiento	30%	no	✖ Detalles

Insertar

Seleccione el Tipo de Indicador a Insertar:

Cuantitativo Cualitativo

Nombre

Descripción

Observación

Plan

Real

Criterio Cumplimiento (%)

Insertar



Modificar

Nombre Indicador: Progreso del Levantamiento

Descripción: Indica el progreso en el levantamiento realizado a cada uno de los 10 departamentos.

Observación

Plan: 10

Real: 3

Criterio Cumplimiento: 100 %

Modificar



ANEXO 6 Prototipo De Caso de Uso Gestionar Tarea.

Listar

Cumplimiento	Nombre de la Tarea	Flujo	Fase	Etapa	Insertar Tarea
Pendiente	Levantamiento de Computadoras	Evaluación	Migracion a Software Libre	Preparación	Indicadores

Insertar

Responsable:

Nombre Tarea:

Flujo:

Descripción:

Fecha Inicio:

Fecha Terminación:

Modificar

[Indicadores](#) [Trabajadores Vinculados](#) [Recursos](#)

Responsable:

Nombre Tarea:

Flujo:

Descripción:

Fecha Inicio:

Fecha Terminación:

ANEXO 7 Prototipo De Caso de Uso Gestionar Vinculación de Trabajadores.

Vincular

Trabajadores de la tarea: Levantamiento de Computadoras

Trabajadores	
Eliza Méndez	✖
Jerislandys Marcos Marcos	✖



ANEXO 8 Prototipo De Caso de Uso Gestionar Asignación de Recursos.

Agregar

Recursos de la Tarea: Levantamiento de Computadoras

Recursos	
Un Lapicero y un Modelo	✖



ANEXO 9 Prototipo De Caso de Uso Visualizar Fase.

 Ver Etapas de esta Fase.

Responsable:	<input type="text" value="William Feal"/>
Nombre Fase:	<input type="text" value="Migracion a Software Libre"/>
Descripción:	<input type="text" value="la Metodologia propuesta por el MES no consta de Fases, por consiguiente esta"/>
Fecha Inicio:	<input type="text" value="2008-12-01"/>
Fecha Terminación:	<input type="text" value="2011-04-01"/>

ANEXO 10 Prototipo De Caso de Uso Gestionar Etapa.
Listar

Nombre de la Etapa	Responsable	Insertar Etapa	
Etapas de la Fase: Migracion a Software Libre			
Preparación	Jose Manuel Ruiz Concepcion	✖	✎
Migración Parcial	Osniel Fariñas	✖	✎
Migración Total	Jose Manuel Ruiz Concepcion	✖	✎
Consolidacion	Osniel Fariñas	✖	✎

Insertar



Responsable:

Nombre Etapa:

Descripción:

Fecha Inicio:
 

Fecha Terminación:
 

Modificar

 [Ver Tareas de esta Etapa.](#)

Responsable:

Nombre Etapa:

Descripción:

Fecha Inicio:
 

Fecha Terminación:
 

ANEXO 11 Prototipo De Caso de Uso Gestionar Fase.

Listar

Nombre de la Fase	Responsable	Insertar Nueva
Migracion a Software Libre	William Feal	✕ ✎

Insertar



Responsable:

Nombre Fase:

Descripción:

Fecha Inicio:
 

Fecha Terminación:
 

Modificar

 [Ver Etapas de esta Fase.](#)

Responsable:

Nombre Fase:

Descripción:

Fecha Inicio:
 

Fecha Terminación:
 

ANEXO 12 Prototipo De Caso de Uso Gestionar Personal.

Listar

Carnet de Identidad	Nombre Y Apellidos	Dirección Particular	Insertar Nuevo	
83082909304	Carlos Javier Palacios	Responsable		
63484364663	Eliza Méndez	Secretaria ind	✖	
69786877961	Jerislandys Marcos Marcos	Trabajador	✖	
69783813867	Jose Manuel Ruiz Concepcion	Adm Almacen	✖	
89283845644	Lisder Oscar Perez Diaz	Decano Inf	✖	
64964694694	Mayrelis Pérez Fabelo	Profesora	✖	
86347683763	Osniel Fariñas	Subdirector Docente	✖	
34364364364	Otoniel Guevara Carvajal	secretario docente informatica	✖	
83082909305	William Feal	Responsable finf	✖	

Insertar

Carnet de Identidad:

Nombre:

Correo(Email):

Función:

Dirección:



Modificar

Carnet de Identidad:

Nombre:

Correo(Email):

Función:

Dirección:



ANEXO 13 Prototipo De Caso de Uso Gestionar Flujo.

Listar

Flujo	Insertar Flujo
Evaluación	✘
Diseño	✘
Pilotos	✘
Formación	✘
Implementación	✘
Asistencia Técnica	✘

Insertar

[Regresar](#)

ANEXO 14 Prototipo De Caso de Uso Gestionar Permiso.

Listar

Login	Nombre Y Apellidos	Insertar Nuevo
Bily	William Feal	✘ 
Eliza	Eliza Méndez	✘ 
inf2004102	Carlos Javier Palacios	
Jerislandis	Jerislandys Marcos Marcos	✘ 
Jose	Jose Manuel Ruiz Concepcion	✘ 
LOPD	Lisder Oscar Perez Diaz	✘ 
Osniel	Osniel Fariñas	✘ 

Insertar

Introduzca un Nombre para este Usuario:

Trabajador:

Contraseña

Repetir Contraseña:

Pregunta de Seguridad:

Respuesta:

Modificar

Modificando los datos de Carlos Javier Palacios, si no desea modificar la contraseña deje el campo en blanco.

Login:

Contraseña:

Repetir Contraseña:

Pregunta de Seguridad:

Respuesta:



ANEXO 15 Prototipo De Caso de Uso Generar Reporte Exportable.

De Fase

Seleccione una Fase para ver sus Características:

De Etapa

Seleccione la Fase a que pertenece y a continuación la Etapa cuyas características desea ver:

Fase:

Etapa:

De Tarea

Seleccione la Fase y Etapa a que pertenece la Tarea cuyas características desea ver:

Fase:

Etapa:

Tarea:



ANEXO 16 Prototipo De Caso de Uso Visualizar Responsabilidades.

[Ver en Línea de Tiempo](#)

Cumplimiento	Nombre de la Tarea	Flujo	Fase	Etapa	
Pendiente	Levantamiento de Computadoras	Evaluación	Migracion a Software Libre	Preparación	Indicadores

En Línea de Tiempo

Línea de Tiempo de Tareas Programada					
Tarea	Etapa	Fase	Inicio	Final	2009
Levantamiento de Computadoras	Preparación	Migracion a Software Libre	2008-12-01	2009-04-30	

ANEXO 17 Prototipo De Caso de Uso Modificar Indicador.

Nombre Indicador: Progreso del Levantamiento
 Descripción: Indica el progreso en el levantamiento realizado a cada uno de los 10 departamentos.
 Plan: 10
 Real:
 Criterio Cumplimiento: 100 %



ANEXO 18 Prototipo de los Casos de Usos Realizar Salva de Seguridad al Sistema de Gestión y al Registro de Diagnósticos.

Módulo para la Elaboración de Salvas de Seguridad

Realizar Salva de Seguridad al Sistema de Gestión

Realizar Salva de Seguridad al Registro de Diagnósticos

ANEXO 19 Prototipo De Caso de Uso Consultar Ayuda.

Temas de Ayuda

Introducción

Administración

Gestión

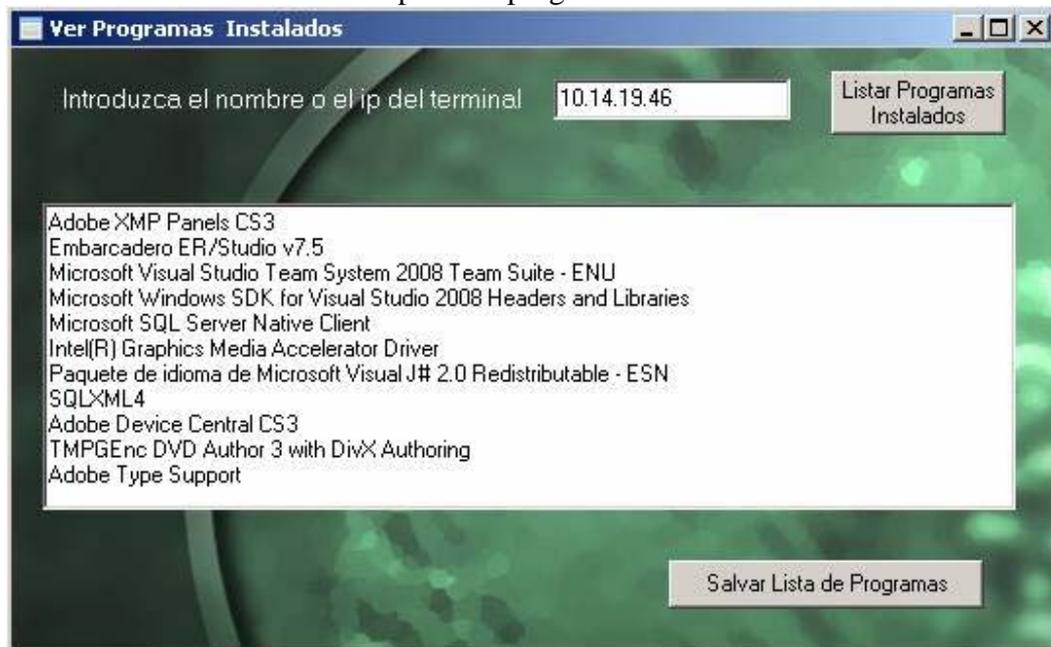
Reportes

Diagnosticar Red

Diagnosticar Equipo

ANEXO 20 Prototipo De Caso de Uso Diagnosticar Equipo.

Utilidad para ver programas instalados.



ANEXO 21 Prototipo De Caso de Uso Descargar Utilidad de Diagnóstico de Equipo.



ANEXO 22 Prototipo De Caso Subir Diagnóstico Realizado a Equipo.

- Analizar los resultados obtenidos por el programa [VerInstalados](#)
- Visualizar los Análisis Previamente Realizados.



ANEXO 23 Prototipo De Caso Eliminar Diagnóstico Innecesario.

- Analizar los resultados obtenidos por el programa [VerInstalados](#)
- Visualizar los Análisis Previamente Realizados.

Ip Equipo	Fecha del Analisis	Hora
10.14.19.46.txt	2009-03-25	16-10-48 ✖



10.14.19.46.txt	2009-03-25	16-11-06 ✖
-----------------	------------	------------

Siguiente-->

ANEXO 24 Prototipo De Caso Visualizar Diagnóstico Realizado a Equipo.

Softwares Libres Conocidos:

AIMP2
 Mozilla Firefox (3.0)
 Picasa 3
 Java(TM) SE Runtime Environment 6
 Java(TM) SE Development KIT 6
 MPlayer for Windows (Full Package)

Otros Softwares Instalados:

Adobe Acrobat 8 Professional - English, Français, Deutsch
 Adobe Flash Player Plugin
 Adobe Photoshop CS3
 Adobe Dreamweaver CS3
 Bejeweled Twist en Español
 Administrador de discos extraíbles de Creative
 Developer Express .NET v8.2
 DVD Shrink 3.2
 Microsoft Office Enterprise 2007
 EVEREST Ultimate Edition v4.60
 FlashGet 1.8.6.1008
 PowerQuest PartitionMagic 8.0
 Kaspersky Anti-Virus 6.0 for Windows Workstations
 K-Lite Mega Codec Pack 4.1.4
 Microsoft .NET Framework 2.0
 Paquete de idioma de Microsoft .NET Framework 2.0 - ESN
 Microsoft Document Explorer 2005
 Paquete de idioma de Microsoft Document Explorer 2005 - Español
 Microsoft SQL Server 2005

ANEXO 25 Prototipo De Caso Diagnosticar Red.

Diagnóstico de los Sistemas Operativos Actuales en los terminales correspondiente al rango de ip 10.14.16.1 hasta 10.14.16.10.
Fecha: 2009-06-16

Para Obtener un Listado de los Softwares Instalados en un terminal con Windows debe usar el programa: [Ver Instalados](#)

Ip	SO/Op.	Detalle
10.14.16.1		Windows
10.14.16.2		Windows
10.14.16.4		Windows
10.14.16.5		Windows
10.14.16.9		Linux

Estadísticas

De 10 ip analizadas, 5 equipos respondieron, de ellos se diagnostican:

4 con Windows
1 con Linux
0 probablemente Linux (Podría tratarse de Router o Switch)

Porcientos
80 % Windows
20 % Linux
0 % Linux???

[Generar Estadísticas](#)