



## **Automatización de la generación del Expediente de Proyecto de Migración**

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

**Autor:**

Javier Eusebio Junquera Falcón

**Tutores:**

Ing. Yasiel Pérez Villazón

Ing. Nurisel Palma Pérez

**Consultantes**

Ing. Mairelys Fernández González

Ing. Marlon Rodríguez García

**La Habana, 2016.**

**“Año 58 de la Revolución”**

## **Declaración de Autoría**

Declaro ser el único autor de este trabajo y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales del mismo, con carácter exclusivo.

Para que así conste firmo la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año 2016.

---

Javier Junquera Falcón

---

Ing. Yasiel Pérez Villazón

---

Ing. Nurisel Palma Pérez



*“Todo lo que una persona pueda imaginarse,  
otras podrán hacerlo realidad”*

*Julio Verne*

## **Dedicatoria**

**A mis padres, por nunca rendirse conmigo**

## Agradecimientos

A mis tutores, gracias a los cuales este trabajo ha sido posible.

A Maire, que me ayudó a salir de más problemas de los que creí que podía encontrar.

A Marlon, que sin su ayuda este trabajo no sería el mismo.

A mis amigos, por soportarme incluso cuando ni yo mismo lo hacía.

A todos los profesores que me ayudaron durante la carrera.

Y más importante a mi familia, que siempre han estado y van a estar dándome su apoyo, incluso cuando no me lo merezco.

## Resumen

Durante la migración de una entidad, se guarda parte de la información necesaria para el desarrollo de la misma en la Plataforma Cubana de Migración a Código Abierto. Debido a la necesidad de gestionar toda la información generada durante todo este proceso, esta se almacena en el Expediente de Proyecto de Migración. El siguiente trabajo de diploma describe los resultados de una investigación cuyo objetivo es desarrollar un componente dentro del subsistema de Migración de la plataforma que posibilite disminuir el tiempo en la generación de los artefactos del expediente. Esto es necesario debido a que al realizarse el mismo de forma manual, se emplea tiempo y esfuerzo extra por parte de los especialistas de migración, surgen errores en la información del expediente como duplicación o falta de la misma. Durante la investigación se realizó un estudio de la información que se tiene en la plataforma y lo que se debe documentar en el expediente, para obtener los artefactos que se pueden generar de forma automática. Se diseñó e implementó un componente capaz de generar estos artefactos mediante el uso de servicios brindados por la plataforma y se le realizaron pruebas que garantizaron el correcto funcionamiento del mismo. Finalmente se obtuvo un componente que facilita la generación de varios artefactos con mayor rapidez y evitando errores que existían anteriormente, logrando así que los especialistas de migración no tengan que emplear tiempo extra en la realización del expediente durante el proceso de migración.

**Palabras Clave:** artefactos, expediente de proyecto, migración, plataforma.

# Índice

Introducción.....	1
Capítulo 1. Fundamentación Teórica.....	5
1.1 Conceptos importantes.....	5
1.2 Plataforma Cubana de Migración a Código Abierto.....	6
1.2.1 Subsistemas de la PCMCA.....	6
1.2.2 Servicios.....	9
1.3 Expediente de Proyecto de Migración.....	10
1.4 Correspondencia entre el Expediente de Proyecto de Migración y la PCMCA.....	14
1.5 Herramientas y Tecnologías.....	24
1.6 Metodología de desarrollo de software.....	26
1.7 Conclusiones parciales.....	27
Capítulo 2. Análisis y diseño.....	28
2.1 Propuesta de la solución.....	28
2.2 Especificación de requisitos de software.....	28
2.3 Historias de Usuario.....	30
2.4 Arquitectura de software.....	37
2.5 Patrones de diseño.....	39
2.6 Conclusiones parciales.....	40
Capítulo 3. Implementación y pruebas.....	41
3.1 Implementación.....	41
3.1.1 Métodos implementados.....	41
3.1.2 Estándar de codificación.....	48
3.1.3 Cambios en el requisito funcional 2.....	50
3.2 Definición de las pruebas.....	51
3.2.1 Pruebas internas.....	52
3.2.2 Pruebas de aceptación.....	59
3.2.3 Pruebas de satisfacción.....	61

3.3 Conclusiones parciales.....	63
Conclusiones.....	64
Recomendaciones.....	65
Bibliografía.....	66
Referencias Bibliográficas.....	67
Anexos.....	69



## Introducción

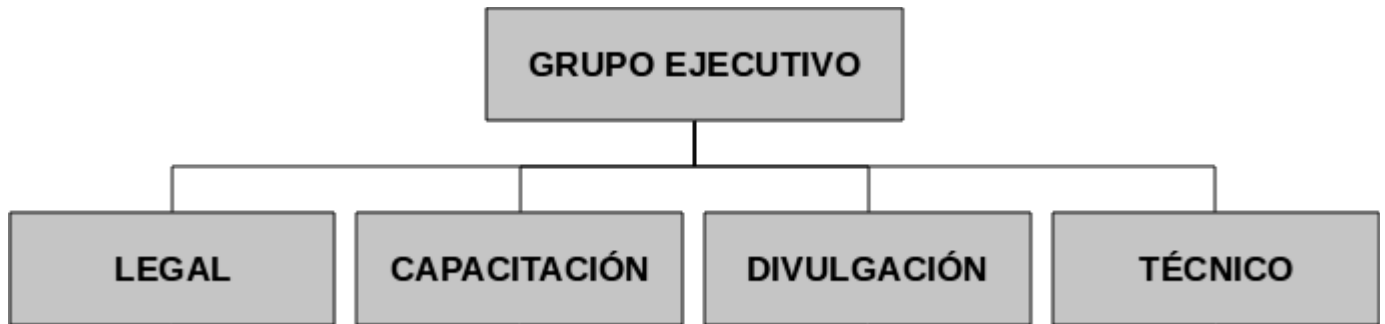
La constante evolución de las tecnologías, unida a la interconexión y globalización de la economía, convierte a las Tecnologías de la Información y las Comunicaciones (TIC) en un elemento estratégico para el crecimiento, maduración y transformación de las organizaciones. El avance de las TIC en el mundo genera un alto nivel de desarrollo y competitividad entre las empresas. Estas, para mantenerse en el mercado, centran sus esfuerzos en optimizar la gestión de la información y en consecuencia adaptan sus objetivos estratégicos. Para mejorar la calidad de los servicios que brindan, la toma de decisiones en las empresas es un elemento indispensable, por lo que es necesario recopilar la información de su funcionamiento a través de reportes y permitir a sus procesos ser representados de una manera sintética.

En la actualidad uno de los pilares fundamentales en el desarrollo de la economía de los países desarrollados, es el vínculo de las universidades con las empresas de bienes y servicios. Los países subdesarrollados, aunque en menor medida, también priorizan este enlace para lograr avances en el desarrollo de su economía. Las empresas cubanas no están ajenas a este proceso, ya que se cuenta con centros productivos en la Universidad de las Ciencias Informáticas (UCI) encargados de potenciar el desarrollo de las TIC en estas empresas.

En el año 2004 se toma el acuerdo 084 del Consejo de Ministros de la República de Cuba, el cual orienta la migración paulatina hacia aplicaciones de código abierto. La migración a software<sup>1</sup> libre es una necesidad en Cuba ya que la mayoría del software usado es de carácter privativo, muchos de los cuales pertenecen a empresas norteamericanas que disponen de la autorización para espiar y recolectar información de los usuarios. Se ocasionaría además un gasto de capital para el país si se tuvieran que pagar las licencias de cada uno de estos software. El proceso de migración es dirigido por el Grupo Ejecutivo, el cual dirige los grupos: Capacitación, Legal, Divulgación y el Grupo Técnico Nacional, como se ilustra en la Figura 1. Las tareas de este último grupo las desarrolla el Centro de Software Libre (CESOL) de la UCI, al cual pertenece el departamento Servicios Integrales de Migración, Asesoría y Soporte (SIMAYS), que es el departamento encargado del proceso de migración.

---

1 *Software: conjunto de programas, instrucciones y reglas informáticas que permiten ejecutar distintas tareas en una computadora.*



*Figura 1: Grupos para la migración a plataformas de código abierto en Cuba*

En este departamento se tienen trazadas líneas de investigación orientadas al desarrollo de metodologías, planes, estrategias y aplicaciones de apoyo al proceso de migración. Como parte del proceso de la informatización de la sociedad cubana en el 2007 se crea la Metodología Cubana para la Migración a Código Abierto (MCMCA) la cual describe las estrategias que se llevan a cabo en los procesos de migración.

Para organizar la documentación que genera este proceso, se decidió crear un grupo de artefactos que conforman el Expediente de Proyecto de Migración como parte de un trabajo de diploma llevado a cabo por la Ing. Anadalis Pérez Rodríguez. También para automatizar el procesamiento de la información se desarrolló la Plataforma Cubana de Migración a Código Abierto (PCMCA) la cual contiene un conjunto de subsistemas que intercambian datos entre sí a través de una capa de servicios web.

Desde el año 2010 SIMAYS se encarga del proceso de migración de diferentes empresas de bienes y servicios en el país, las cuales pertenecen a varios OACE que se encuentran en el proceso de migración. Entre estas empresas se encuentra la Empresa Telemática del Ministerio de la Industria Pesquera (TELEMAR). En el año 2012 se comienza la migración del Centro de Cibernética Aplicada a la Medicina (CECAM). En el año 2013 se migra la Empresa Constructora de Obras de Arquitectura #24 (ECOA 24) y en el 2014 la Empresa Constructora de Obras de Arquitectura e Industriales (ECOAIND 3).

En el proceso de migración se crea un puesto de mando donde se centraliza toda la información correspondiente a cada una de estas entidades. En este, se instala la PCMCA en una computadora, donde hay una persona encargada de introducir toda la información que los especialistas que ejecutan la migración, traen al finalizar la tarde. Todos los días se ejecutan estas acciones hasta que al concluir la

migración, los especialistas confeccionan de forma manual y en horarios fuera de la jornada laboral, los artefactos del Expediente de Proyecto de Migración. Esta tarea trae consigo empleo de tiempo y esfuerzo extra, así como duplicación de la información y errores humanos.

Una vez planteado lo anterior, se identifica el siguiente **problema científico**: ¿Cómo disminuir el tiempo en la realización de artefactos del Expediente de Proyecto de Migración?

El **objeto de estudio** del presente trabajo se centra en la Plataforma Cubana de Migración a Código Abierto, enmarcado en el **campo de acción** el subsistema de Migración de la PCMCA. Como **objetivo general** se plantea: Desarrollar un componente dentro del subsistema de Migración de la PCMCA que posibilite disminuir el tiempo en la realización de los artefactos del Expediente de Proyecto de Migración.

Como **objetivos específicos** se establecieron los siguientes:

- Realizar un estudio de la información existente en la PCMCA y la que se debe documentar en los artefactos del Expediente de Proyecto de Migración.
- Desarrollar el componente que posibilite disminuir el tiempo en la realización de los artefactos del Expediente de Proyecto de Migración en el subsistema de Migración de la PCMCA.
- Probar el componente implementado.

Para dar cumplimiento a los objetivos anteriores se establecieron las siguientes **tareas de investigación**:

- Estudio y análisis del subsistema de Migración de la PCMCA.
- Análisis del Expediente de Proyecto de Migración.
- Definición de los requisitos funcionales y no funcionales a implementar.
- Definición de la arquitectura del componente de generación del Expediente de Proyecto de Migración.
- Implementación de un componente que posibilite la automatización de la generación de los artefactos del Expediente de Proyecto de Migración en el subsistema de Migración de la PCMCA.
- Realización de pruebas al componente implementado.

Por tanto se plantea lo siguiente como **idea a defender**: El desarrollo de un componente dentro del subsistema de Migración de la PCMCA hará posible disminuir el tiempo en la realización de los artefactos del Expediente de Proyecto de Migración por el equipo de especialistas de migración.

Para el desarrollo del proceso de investigación se empleó un método teórico y una técnica de recopilación de información, que a continuación se detallan:

El **método analítico-sintético** consiste en la extracción de las partes de un todo, con el objetivo de estudiarlas y examinarlas por separado para luego sintetizarlas y tomar los elementos más importantes. Este método fue usado para el análisis de la PCMCA y sus subsistemas, de los cuales es necesario extraer información pertinente para la creación del Expediente de Proyecto de Migración. Además se empleó para analizar los distintos artefactos que componen el expediente, para conocer la información requerida por cada uno de estos y determinar cuáles serán generados desde la PCMCA.

La técnica **entrevista** propició la obtención de información a partir de la experiencia de personas involucradas en el proceso de migración de los OACE. Esta se empleó para conocer detalles de los problemas que han existido en este proceso a la hora de generar el Expediente de Proyecto de Migración de forma manual.

El presente documento se encuentra dividido en tres capítulos, en el primero de ellos, **Fundamentación teórica**, se abordan los conceptos más importantes del tema así como una descripción de la PCMCA y sus subsistemas. Se realiza un análisis de los artefactos del Expediente de Proyecto de Migración y se definen los que se generan a partir de la plataforma. Se explican las distintas herramientas y tecnologías que se van a usar, así como la metodología de desarrollo de software.

En el segundo capítulo, **Análisis y diseño**, se realiza una propuesta de la solución y se definen los requisitos funcionales y no funcionales de la misma. Además se describen las funcionalidades en las Historias de Usuario (HU). Se explican los patrones arquitectónicos que se van a usar y se define la arquitectura de software.

En el capítulo 3, **Implementación y pruebas**, se definen los estándares de codificación del lenguaje usados en la implementación. Se definen y realizan casos de pruebas para cada uno de los tipos de pruebas definidos para comprobar el correcto funcionamiento del componente implementado. Se obtienen los resultados de las pruebas y se verifica que se cumplen los objetivos planteados al inicio del documento.

# Capítulo 1. Fundamentación Teórica

En este capítulo se definen algunos conceptos de términos usados en la investigación. Se hace un estudio de los distintos subsistemas de la PCMCA y del Expediente de Proyecto de Migración, con el objetivo de determinar su correspondencia. Además se definen las herramientas y tecnologías a utilizar y se expone la metodología de desarrollo de software.

## 1.1 Conceptos importantes

En el desarrollo de esta investigación, se usan a menudo términos cuyo entendimiento puede resultar difícil para aquellos no familiarizados con la materia, como migración, expediente de proyecto y artefacto, los cuales se definen a continuación.

Para el desarrollo de esta investigación se define **migración**, según la Guía cubana de migración de aplicaciones de código abierto, como: *"un proceso ordenado en el cual se sustituye, parcial o totalmente, el software existente en la organización por alternativas liberadas bajo licencias libres o de código abierto [1]"*.

Basándose en la definición de **expediente de proyecto** de la Ing. Anadalis Pérez Rodríguez se entiende este como *"el recurso que permite registrar el ciclo de vida de un proyecto y que garantiza recoger la información necesaria e importante, permitiendo así una reutilización de la misma. Teniendo como característica fundamental brindar una estructura que proporcione organización y que posibilite el control de cada uno de los artefactos generados, sin que la información se repita innecesariamente, contribuyendo a que la dedicación y el esfuerzo de los especialistas no sean excesivos en el proceso de documentación [2]"*.

Partiendo de los anteriores conceptos, el autor de la investigación plantea que un **Expediente de Proyecto de Migración** *"es un conjunto de documentos que reúne los datos registrados durante el seguimiento y hasta la finalización del proceso de migración en una entidad"*.

Durante el transcurso de esta investigación se define como **artefacto**, basándose en la definición de la Ing. Anadalis Pérez Rodríguez, *"cualquier producto medible y verificable que se elabora para documentar un proceso o parte de uno. Estos ayudan a definir el alcance del proyecto y a controlar el avance del"*

trabajo” [2].

## **1.2 Plataforma Cubana de Migración a Código Abierto**

La PCMCA es una aplicación web que automatiza la gestión de la información generada en el proceso de migración a plataformas de código abierto de las empresas e instituciones. Cuenta con subsistemas que interactúan entre sí en un esquema de servicios, compartiendo datos y funcionalidades. Comparten una interfaz única de fácil manejo y usabilidad para el usuario, que permite la gestión de los procesos de migración y el acceso a las distintas aplicaciones de la plataforma. A continuación se describen los distintos subsistemas y servicios que componen la PCMCA.

### **1.2.1 Subsistemas de la PCMCA**

- **Directorio de aplicaciones**

Permite el almacenamiento de perfiles de aplicaciones informáticas, relacionándolas por funcionalidades similares. Su función más importante es proponer alternativas libres o de código abierto para aplicaciones privativas previamente establecidas o a partir de determinadas características.

- **Certificación de hardware**

Permite el almacenamiento de perfiles de hardware. Su función más importante es la certificación del hardware encontrado en las empresas el cual puede desencadenar el proceso de homologación en el caso de que no existan controladores válidos para el hardware que se está certificando, siempre y cuando la empresa cuente con los recursos necesarios para adquirir un hardware homólogo al que pretendía certificar.

- **Planificación**

Consiste en una aplicación para el control de proyectos de migración, y toda la información relativa a los mismos en lo que se refiere a tareas, participantes, recursos y entidades. El sistema permite además, planificar el proceso de migración y controlar su avance a partir de indicadores.

- **Inventario**

Permite realizar el inventario de activos informáticos, teniendo en cuenta las aplicaciones y los dispositivos de hardware que se encuentran en un área determinada, este subsistema depende de la información recopilada por el OCSInventory<sup>2</sup> el cual realiza el escaneo de los ordenadores y guarda la información en una base de datos.

- **Encuestas**

Permite la gestión de encuestas teniendo en cuenta los perfiles de aplicaciones asociados a cada proyecto de migración.

- **Seguridad**

Se encarga de validar los privilegios de los usuarios y el acceso a los sistemas que conforman la PCMCA.

- **Soporte**

El subsistema de soporte ofrece la posibilidad de gestionar incidencias que tienen los usuarios durante el proceso de migración lo cual es posible a través de la planificación del soporte durante este período.

- **Migración**

El subsistema de Migración es un componente, que mediante su relación con los demás subsistemas de la PCMCA permite la realización automatizada de muchas de las actividades orientadas por la Metodología Cubana de Migración, mediante la gestión y análisis de la información que se genera durante todo el proceso de migración [3]. El submenú de este subsistema contiene cinco elementos:

- Proyectos, donde se guardan todos los datos referentes a los proyectos de migración existentes en la PCMCA.
- Perfiles, en el que se listan las distintas funciones de las aplicaciones para facilitar la migración.
- Capacitación, que contiene los planes de capacitación a usar durante el proceso de migración.
- Servicios de red, que muestra los tipos de servicios de redes existentes, lo cual permite una mayor facilidad a la hora de migrar los servicios.

---

<sup>2</sup> *OCSInventory: Es un software libre que permite recopilar información de los ordenadores en los que se encuentra y manejar el inventario de archivos informáticos que crea.*

- Reglas, donde se permite encontrar una solución libre para cada sistema operativo de carácter privativo.

En la sección Proyectos se permite crear un proyecto distinto para cada una de las entidades que se están migrando y guardar en cada uno de estos proyectos toda la información necesaria para realizar el proceso de migración. Cuando se selecciona uno de estos proyectos, del mismo se ofrece (ver Figura 2):

- Información general, que contiene un resumen de la información del proyecto.
- Estado de las actividades, donde se listan las actividades a llevar a cabo durante la realización de este proyecto.
- Participantes, que ofrece un listado de las personas que participan en el proceso de migración.
- Aplicaciones, que contiene un listado con las aplicaciones existentes en la entidad y las soluciones que se proponen para cada una.
- Computadoras, donde se listan las computadoras registradas a la entidad, los datos de cada una y los dispositivos de hardware que estas poseen.
- Redes, que contiene la información de la red de la entidad.
- Estado de las tareas, que muestra un diagrama con las tareas a realizar y el estado en el que estas se encuentran.

Una vez analizados los subsistemas de la PCMCA se puede concluir que, a pesar de que cada uno de ellos brinda información necesaria para el proceso de migración, esta se encuentra de forma muy generalizada. Sin embargo en el subsistema de Migración, se agrupa toda esta información y más importante, se define a qué proyecto pertenece la misma. Por tanto se determina que se usará la información brindada por este subsistema para la generación de los artefactos del Expediente de Proyecto de Migración.



Información general	
Nombre del proyecto	ECO A
Fecha de inicio	2015-05-16
Fecha de fin	2015-08-31
Objetivo general	Lograr migrar toda la institución
Prioridad	alto
Modalidad	Foraneo
Responsable por parte del cliente	Manuel Alejandro
Conexión con el subsistema inventario	<a href="#">Conexión creada</a> <a href="#">Configurar</a> <a href="#">Detalles de un perfil seleccionado.</a> <a href="#">Datos</a>
Conexión con el subsistema encuesta	<a href="#">No existe una conexión creada</a> <a href="#">Configurar</a>
Descripción del proyecto	Proyecto de migración a código abierto

Figura 2: Submenú Proyectos del subsistema de Migración

La obtención de esta información existente en el subsistema de Migración es posible mediante el uso de varios servicios ofrecidos en la PCMCA, los cuales serán descritos a continuación.

### 1.2.2 Servicios

La PCMCA ofrece una gran variedad de servicios web que realizan diferentes funciones en dependencia del subsistema al que pertenecen. A continuación se detallarán algunos de los servicios que ofrece el subsistema de Migración, los cuales son funcionalidades implementadas para obtener los datos de las entidades existentes.

1. `listarPCs($_SESSION['apm']['idproyecto'])`: este servicio identifica el proyecto en el que se está trabajando y devuelve un listado con todas las PCs que pertenecen a dicho proyecto.
2. `listarHardwarePCByIdsPC()`: este servicio devuelve la cantidad de hardware que se encuentra en una computadora en específico y una lista con cada uno de estos dispositivos.
3. `listarAplicacionComputadora()`: dada una computadora, se obtienen la cantidad de aplicaciones en la misma y una lista con todas estas.
4. `listarDispositivosHomologadosByPCs()`: este servicio devuelve todos los dispositivos de hardware homologados por cada PC.

5. `listarServicioRed()`: con este se pueden obtener todos los servicios telemáticos de la entidad.
6. `listarProyectoByIdsPCS()`: dado el identificador de un proyecto devuelve todas las características del mismo.

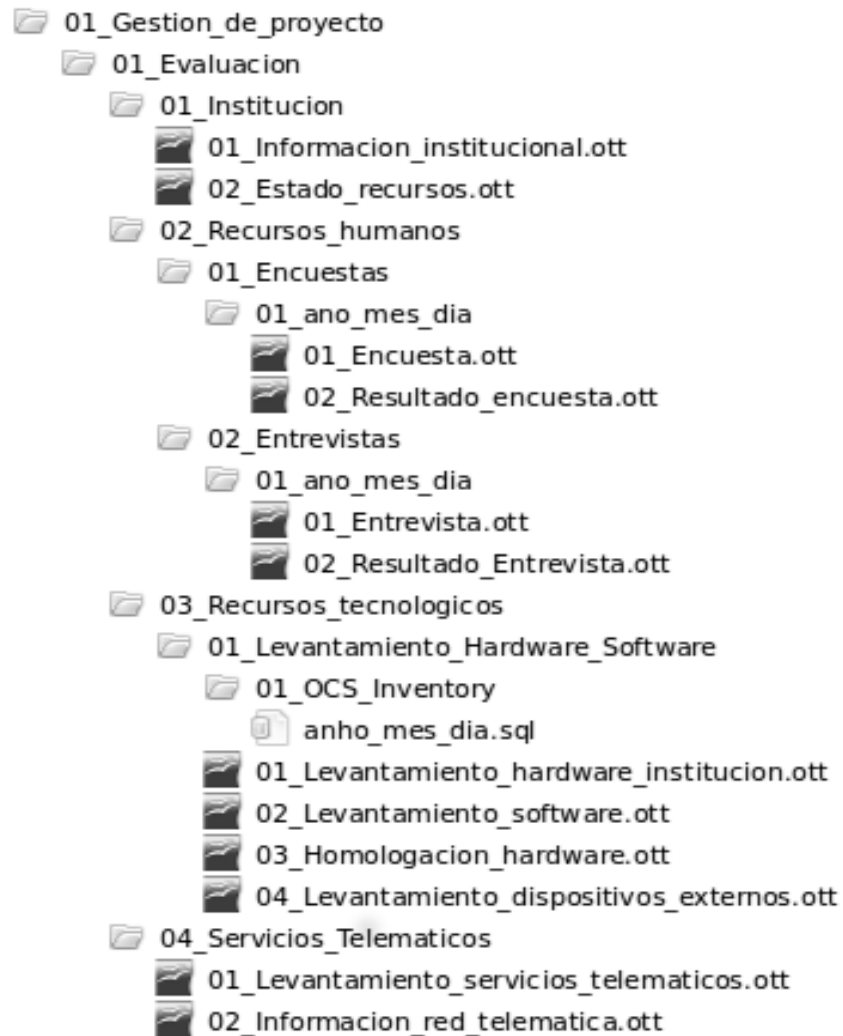
Una vez conocida la forma de obtener la información de la PCMCA, se realizará un estudio del Expediente de Proyecto de Migración para conocer los artefactos cuya información puede ser obtenida en la plataforma.

### **1.3 Expediente de Proyecto de Migración**

El Expediente de Proyecto de Migración está compuesto por 33 artefactos que recogen toda la información de la empresa que se está migrando, necesaria para el departamento encargado de realizar dicho proceso. El expediente contiene dos directorios que serán detallados a continuación:

- **01\_Gestión\_de\_proyecto:** Agrupa los directorios que representan los flujos principales de la metodología, mediante los cuales los especialistas organizan y administran los recursos que intervienen en el proceso de migración de manera tal que este se pueda culminar dentro del alcance, del tiempo y del coste definidos [2].

Este directorio contiene dentro el directorio **01\_Evaluación**, el cual contiene a su vez los directorios **01\_Institución**, donde se encuentran los artefactos que recogen la información acerca de la empresa que se está migrando y los recursos de los que dispone la misma; **02\_Recursos\_humanos**, donde se guardan los artefactos con la información acerca de las encuestas y entrevistas realizadas al personal de la empresa; **03\_Recursos\_tecnológicos**, donde están los artefactos que contienen la información de todos los software y los dispositivos de hardware de la entidad y **04\_Servicios\_Telemáticos**, donde se encuentran los artefactos que contienen la información de los servicios de red que existen en la empresa (ver Figura 3).



*Figura 3: Gestión de Proyecto: Evaluación*

Otro de los directorios dentro de **01\_Gestión\_de\_proyecto** es **02\_Diseño**, el cual a su vez contiene los directorios **01\_Económico**, donde se guarda el artefacto referente al estudio de viabilidad económica realizada a la empresa; **02\_Sensibilización**, donde se encuentra el artefacto con la estrategia de sensibilización diseñada para el personal de la empresa; **03\_Capacitación**, que posee el artefacto con la estrategia de capacitación que se le va a impartir al personal de la entidad; **04\_Migración\_sistema\_operativo**, con los artefactos con la información acerca de los

cambios que se le van a hacer a los sistemas operativos detectados en la empresa; **05\_Migración\_servicios\_telemáticos**, dentro del cual están los artefactos con la información de los cambios necesarios para migrar los sistemas operativos; **06\_Riesgos**, que contiene el artefacto con el plan para la gestión de los riesgos existentes en el proceso de migración y **07\_Plan\_de\_migración**, que guarda el artefacto donde se encuentra definido el plan de migración para realizar este proceso en la entidad. El tercer directorio dentro de **01\_Gestión\_de\_proyecto** es **03\_Pilotos**, dentro del cual se encuentra el directorio **01\_Pruebas** que a su vez contiene el artefacto donde se detallan las pruebas a realizar al sistema operativo (ver Figura 4).

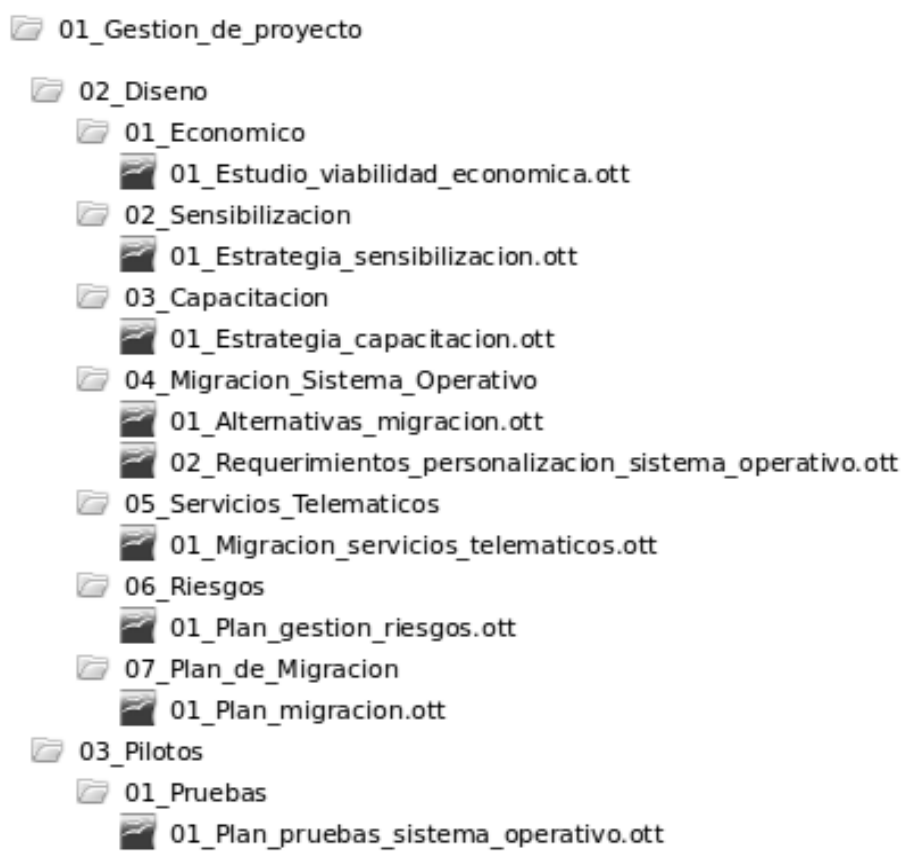


Figura 4: Gestión de Proyecto: Diseño y Pilotos

**01\_Gestión\_de\_proyecto** también posee el directorio **04\_Capacitación**, donde se guardan los

documentos que detallan los cursos a impartir a los usuarios que usarán los nuevos software una vez se termine la migración. Además está el directorio **05\_Implementación**, que contiene los artefactos con la información general del proceso de migración y el directorio **06\_Soporte\_Técnico**, en el cual está el artefacto con la información de la estrategia definida para brindar soporte a la entidad durante el proceso de migración y al terminar este (ver Figura 5).

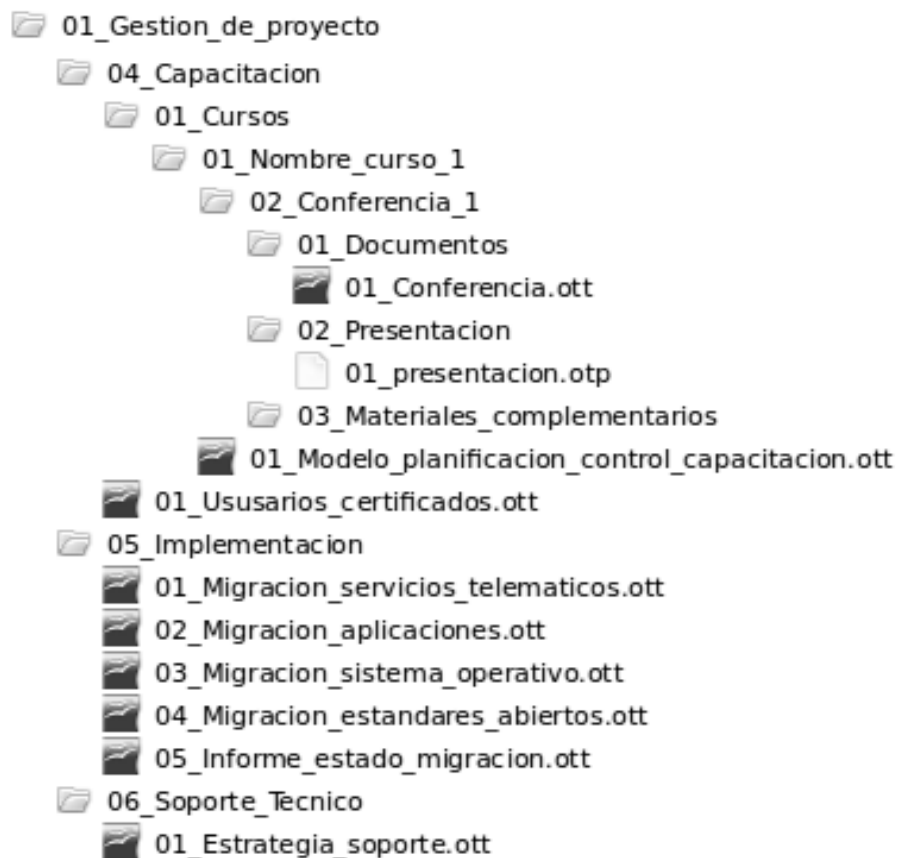


Figura 5: Gestión de Proyecto: Capacitación, Implementación y Soporte Técnico

- **02\_Legal:** Este directorio contiene los artefactos generados por los flujos de apoyo a la migración en dos directorios **01\_Concepción\_del\_proyecto** y **02\_Cierre\_del\_proyecto** y un tercer directorio: **03\_Minuta\_de\_reunión**, en el que se almacenarán las minutas de reunión que se realicen en cada uno de los encuentros entre las partes como constancia legal de las mismas (ver

Figura 6). Estos artefactos siempre son para temas legales, para aclarar los puntos comunes entre clientes y proveedores. Recoge todos los documentos que son necesarios para la entrega de los productos, y su comercialización en el mercado [2].

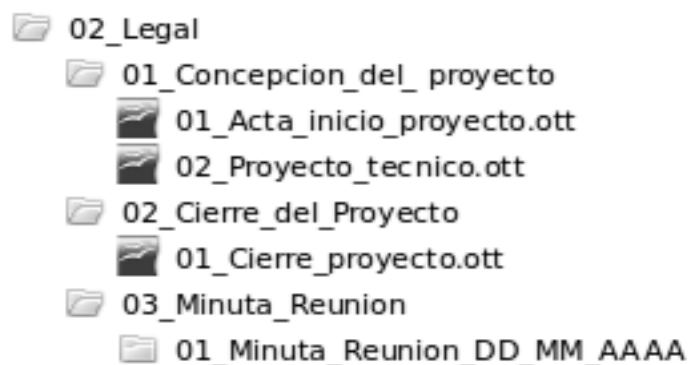


Figura 6: Legal

#### 1.4 Correspondencia entre el Expediente de Proyecto de Migración y la PCMCA

Para dar cumplimiento al primer objetivo específico de la presente investigación, se realizó un análisis de la información en común entre la ofrecida por la PCMCA y la requerida por los artefactos del expediente. Al comparar esta información, se obtienen 10 artefactos cuyo contenido coincide con aquel que se puede extraer directamente de la plataforma. Estos artefactos se describen a continuación:

1. **Levantamiento hardware:** En este documento se muestran los datos del hardware de la entidad de manera general en una tabla y en otra tabla de manera específica para cada una de las computadoras que radican en ella. Para generar este artefacto, en la primera tabla se guardarán los datos de los distintos dispositivos de hardware creando una fila nueva por cada dispositivo existente de un tipo determinado y el total de dispositivos de la entidad (Tabla 1). En la segunda tabla quedan recogidas las características de los recursos de cada una de las computadoras de la institución, especificando el responsable de cada una de ellas (Tabla 2). Esta última debe duplicarse por cada persona que tenga una computadora bajo su responsabilidad.

El artefacto se encuentra en Gestión de proyecto/ Evaluación/ Recursos tecnológicos/ Levantamiento de hardware y software/. La información necesaria para su generación se encuentra dentro del subsistema de Migración, donde para cada uno de los proyectos existentes

se tiene un listado de computadoras las cuales contienen a su vez un listado con todos los dispositivos de hardware de cada una de ellas y su información (ver Figura 7).

*Tabla 1: Recursos de Hardware*

<b>Recursos de la entidad</b>			
<b>Hardware de almacenamiento</b>			
Modelo	Capacidad	Cantidad	
<b>Bios</b>			
Fabricante	Modelo	Cantidad	
<b>Memoria RAM</b>			
Capacidad	Velocidad	Tipo	Cantidad
<b>Monitor</b>			
Fabricante	Modelo	Tipo	Cantidad
<b>Procesador</b>			
Tipo	Velocidad	Cantidad de procesadores	Cantidad
<b>Red</b>			
Velocidad	Modelo		Cantidad
<b>Tarjeta sonido</b>			
Fabricante			Cantidad
<b>Tarjeta de video</b>			
Modelo	Memoria	Cantidad	
<b>Total de dispositivos:</b>			

Tabla 2: Especificación por computadora

Nombre y Apellidos del Responsable:		
Hardware de almacenamiento		
Modelo	Capacidad	
Bios		
Fabricante	Modelo	
Memoria RAM		
Capacidad	Velocidad	Tipo
Monitor		
Fabricante	Modelo	Tipo
Procesador		
Tipo	Velocidad	Cantidad de procesadores
Red		
Velocidad	Modelo	
Tarjeta sonido		
Fabricante		
Tarjeta de video		
Modelo	Memoria	



Figura 7: Lista de dispositivos de Hardware

Total de dispositivos de hardware existentes: 13				
	Tipo	Datos	Permitido	Certificado
<input type="checkbox"/>	Audio	No Certificado. Información original: Fabricante: Intel   Nombre: Intel(r) 82801AA AC'97 Audio Controller   Descripción: Intel(r) 82801AA AC'97 Audio Controller	Si	No
<input type="checkbox"/>	Disco	No Certificado. Información original: Fabricante: (Standard disk drives)   Nombre: VBOX HARDDISK   Modelo: //./PHYSICALDRIVE0   Tipo: Fixed hard disk media   Capacidad: 20473MB	Si	No
<input type="checkbox"/>	Disco	No Certificado. Información original: Fabricante: (Standard CD-ROM drives)   Nombre: VBOX CD-ROM   Modelo: VBOX CD-ROM   Tipo: CD-ROM   Capacidad: 598MB	Si	No
<input type="checkbox"/>	Disco	No Certificado. Información original: Fabricante: (Standard disk drives)   Nombre: VBOX HARDDISK   Modelo: //./PHYSICALDRIVE0   Tipo: Fixed hard disk media   Capacidad: 10236MB	Si	No
<input type="checkbox"/>	Disco	No Certificado. Información original: Fabricante: (Standard CD-ROM drives)   Nombre: VBOX CD-ROM   Modelo: VBOX CD-ROM   Tipo: CD-ROM   Capacidad: 0MB	Si	No

Figura 7: Lista de dispositivos de Hardware

**2. Levantamiento software:** Este documento deja constancia tanto de las aplicaciones libres como de las privativas instaladas en la empresa, si es libre se busca la mejor alternativa libre a esa que es usada por la institución, si es privativa se busca entonces la alternativa libre correspondiente. Para guardar esta información se emplean tres tablas. En la primera tabla se especifican las aplicaciones que están instaladas en la empresa, ya sean libres o propietarias, especificando sus características (Tabla 3). En la segunda se listan las aplicaciones propias usadas en la institución y que han sido desarrolladas con su esfuerzo propio o no y que son utilizadas internamente para prestar servicios en la entidad, estas son una selección de la tabla anterior (Tabla 4). En la última se listan las aplicaciones críticas usadas en la institución, sin las cuales el correcto funcionamiento de la empresa se ve afectado (Tabla 5). En cada una de estas tablas se creará una fila nueva por cada aplicación existente que cumpla con las condiciones de cada tabla. Este artefacto se encuentra en Gestión de proyecto/ Evaluación/ Recursos tecnológicos/ Levantamiento de hardware y software/. La información necesaria para la generación del mismo se puede encontrar en el subsistema de Migración donde para cada uno de los proyectos existentes

se tiene un listado con todos los software instalados en las computadoras de la entidad y la información de los mismos (ver Figura 8). Para la Tabla 3 solo será posible obtener de la PCMCA el nombre y la versión de la aplicación. Para las Tablas 4 y 5 solo se puede obtener el nombre de la aplicación.

*Tabla 3: Aplicaciones Instaladas*

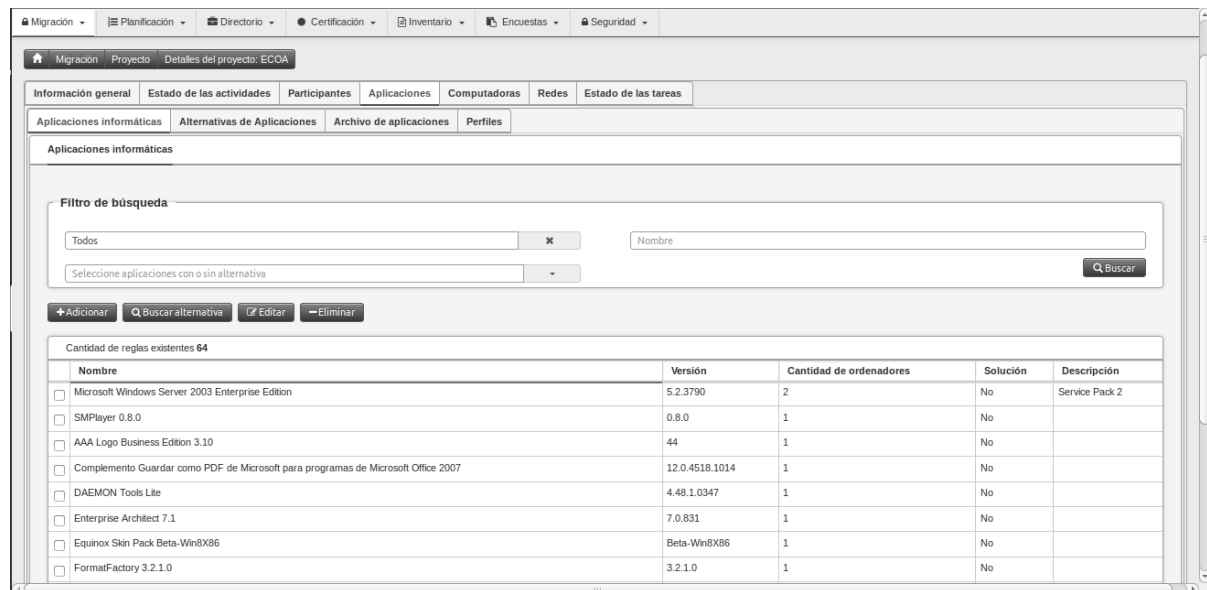
Nombre	Licencia	Versión	Sistemas que la utilizan	Función	SO base
--------	----------	---------	--------------------------	---------	---------

*Tabla 4: Aplicaciones Propias*

Aplicaciones	Función de la aplicación	Descripción
--------------	--------------------------	-------------

*Tabla 5: Aplicaciones Críticas*

Nombre	Función de la aplicación
--------	--------------------------



*Figura 8: Lista de aplicaciones*

**3. Homologacion hardware:** Este documento contiene el listado de hardware y su compatibilidad con el sistema operativo en cuestión a partir del estudio realizado del hardware recogido en el artefacto Levantamiento hardware, así como una descripción de estos y la clasificación (Tabla 6). Para recoger la información en esta tabla se separarán los distintos dispositivos de hardware en internos o externos y se creará una nueva fila por cada dispositivo existente de cada tipo. Este artefacto se encuentra en Gestión de proyecto/ Evaluación/ Recursos tecnológicos/ Levantamiento de hardware y software/. La información necesaria para la generación del mismo se puede encontrar en el subsistema de Migración donde para cada uno de los proyectos existentes se tiene un listado de computadoras las cuales contienen a su vez un listado con todos los dispositivos de hardware de cada una de ellas y su información (ver Figura 7).

*Tabla 6: Dispositivos*

Dispositivos de hardware	Descripción	Clasificación
	Dispositivos internos	
	Dispositivos externos	

**4. Levantamiento dispositivos externos:** Este documento muestra la información sobre los diferentes dispositivos externos de la institución y las características asociadas a ellos. Para recoger esta información existen tres tablas, una para los dispositivos de entrada (Tabla 7), de salida (Tabla 8) o de entrada\salida (Tabla 9). En cada una de estas tablas se creará una fila nueva por cada dispositivo del tipo correspondiente.

Este artefacto se encuentra en Gestión de proyecto/ Evaluación/ Recursos tecnológicos/ Levantamiento de hardware y software/. La información necesaria para la generación del mismo se puede encontrar en el subsistema de Migración donde para cada uno de los proyectos existentes se tiene un listado de computadoras las cuales contienen a su vez un listado con todos los dispositivos de hardware de cada una de ellas y su información (ver Figura 7).

*Tabla 7: Dispositivos de entrada*

Nombre	Marca	Modelo
--------	-------	--------

*Tabla 8: Dispositivos de salida*

Nombre	Marca	Modelo
--------	-------	--------

*Tabla 9: Dispositivos de entrada\salida*

Nombre	Marca	Modelo
--------	-------	--------

- 5. Levantamiento\_servicios\_telematicos:** Este documento tiene la información de los servicios telemáticos que se brindan en la entidad. Para recoger esta información existe una tabla con el nombre, sistema operativo, el software usado por el servicio, la cantidad de usuarios que lo utilizan, cuántos usuarios lo pueden usar a la misma vez y alguna observación acerca de este servicio (Tabla 10). Para cada uno de los servicios existentes en la entidad se creará una fila nueva en la tabla antes mencionada.

Este artefacto se encuentra en Gestión de proyecto/ Evaluación/ Recursos tecnológicos/ Levantamiento de hardware y software/. La información necesaria para la generación del mismo se puede encontrar en el subsistema de Migración donde para cada uno de los proyectos existe una sección de la PCMCA dedicada a la información de las redes de la entidad en la que se pueden encontrar los servicios telemáticos disponibles y la información de los mismos (ver Figura 9).

*Tabla 10: Levantamiento de servicios telemáticos*

Nombre servicio	SO base actual	Software utilizado	Cantidad usuarios	Concurrencia	Observaciones
-----------------	----------------	--------------------	-------------------	--------------	---------------

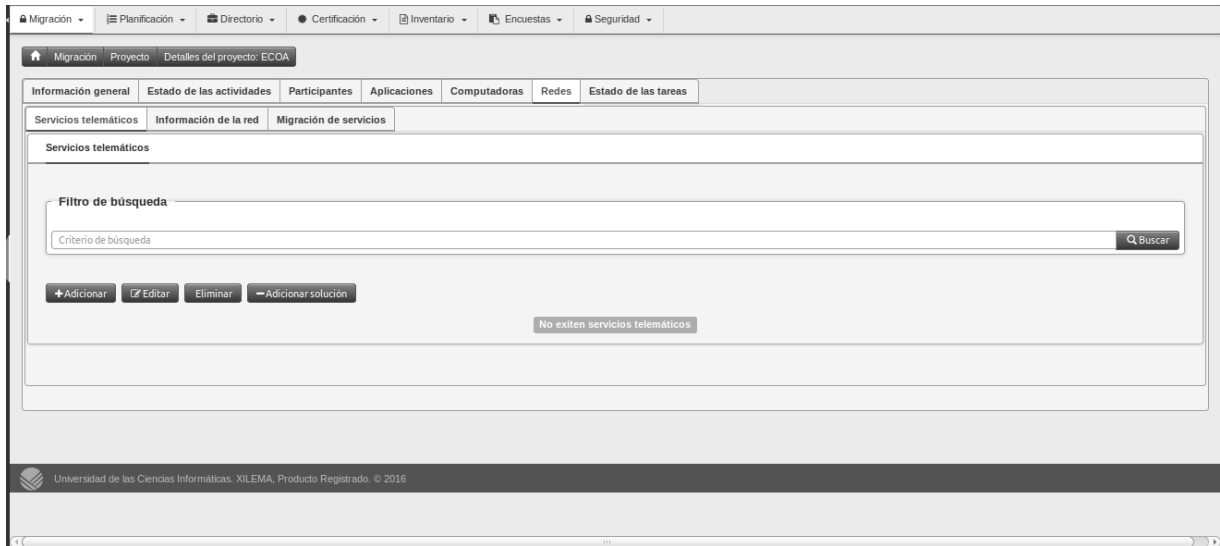


Figura 9: Lista de Servicios Telemáticos

**6. Migracion\_servicios\_telematicos:** En este documento quedan especificados los servicios telemáticos que tiene la institución, tanto los que se utilizan para el trabajo interno como los utilizados para la prestación de servicios. Para ello existe una tabla donde se recoge el nombre de cada servicio, el sistema operativo y el software que usaba inicialmente, el sistema operativo y el software o conjunto de ellos necesarios para que este funcione después de la migración y algunas observaciones acerca de este servicio (Tabla 11). Se creará una fila nueva por cada servicio registrado.

Este artefacto se encuentra en Gestión de proyecto/ Diseño/ Servicios Telemáticos/ Levantamiento de hardware y software/. La información necesaria para la generación del mismo se puede encontrar en el subsistema de Migración donde para cada uno de los proyectos existe una sección de la PCMCA dedicada a la información de las redes de la entidad en la que se puede encontrar la solución dada para la migración de cada uno de los servicios telemáticos existentes (ver Figura 9).

Tabla 11: Servicios telemáticos

Nombre servicio	SO base inicial	Software inicial	SO base final	Software/Conjunto de software final	Observaciones
-----------------	-----------------	------------------	---------------	-------------------------------------	---------------

**7. Migracion\_servicios\_telematicos(2):** Este documento contiene un informe del estado del avance de la migración en las instituciones donde se lleve a cabo el proceso, recogiendo aspectos tanto de las estaciones de trabajo clientes como de los servidores. Para ello se tienen dos tablas, una para mostrar el avance de la migración de los servicios telemáticos de la entidad (Tabla 12) y una para registrar los datos de los servicios migrados (Tabla 13).

Este artefacto se encuentra en Gestión de proyecto/ Implementación/. La información necesaria para la generación del mismo se puede encontrar en el subsistema de Migración, el cual posee, en la sección principal de cada uno de los proyectos existentes en la PCMCA, la información del avance de la migración de los servicios telemáticos (ver Figura 10). Para la segunda tabla se recogen los datos de la sección del subsistema de Migración dedicada a las redes de la entidad (ver Figura 9) y se crea una nueva fila por cada servicio.

Tabla 12: Migración de servicios telemáticos

Servicios telemáticos
Cantidad de servicios telemáticos
Cantidad migrados
Resumen

Tabla 13: Informe de migración de servicios telemáticos

Nombre servicio	Software inicial	Software/Conjunto de Software final	Observaciones
-----------------	------------------	-------------------------------------	---------------

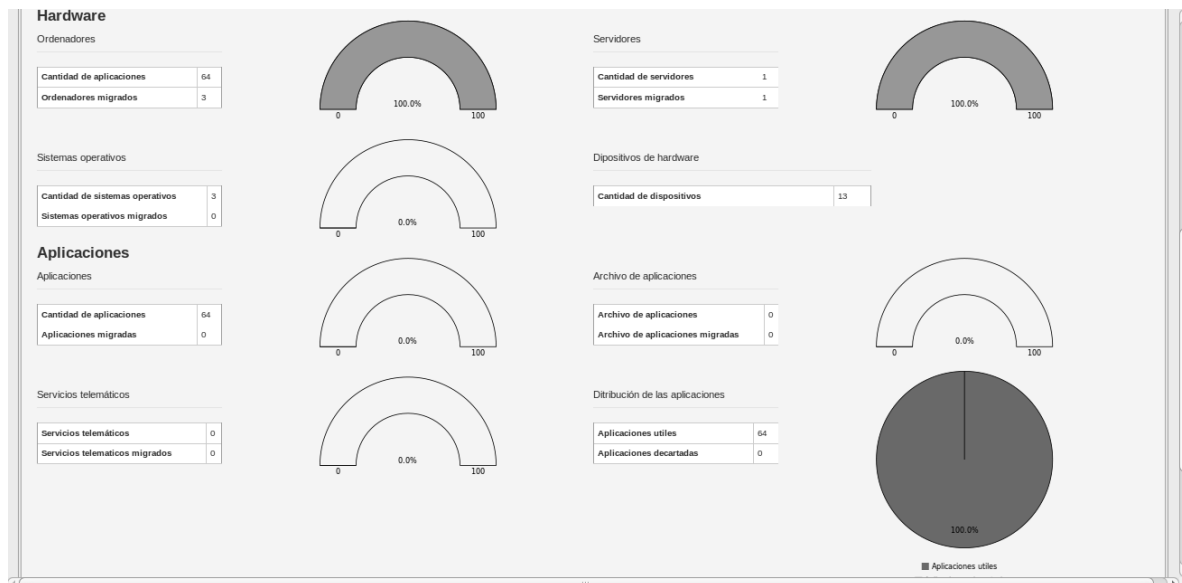


Figura 10: Información general de un proyecto

**8. Migracion\_aplicaciones:** Este documento recoge parte de los datos necesarios para tener un control sobre el avance de la migración. Para ello se tiene una tabla donde se recoge la cantidad de aplicaciones de la entidad, cuántas de estas han sido migradas y qué porcentaje del total representan las aplicaciones migradas (Tabla 13).

Este artefacto se encuentra en Gestión de proyecto/ Implementación/. La información necesaria para la generación del mismo se puede encontrar en el subsistema de Migración, el cual posee, en la sección principal de cada uno de los proyectos existentes en la PCMCA, la información del avance de la migración de las aplicaciones de la entidad (ver Figura 10).

Tabla 14: Migración de aplicaciones

Total de aplicaciones	Migradas	%
-----------------------	----------	---

**9. Migracion\_sistema\_operativo:** Este documento contiene la información del estado de la migración con respecto a la instalación del sistema operativo en las computadoras clientes. Para esta información se tiene una tabla que recoge el total de sistemas operativos en la entidad y

cuántos han sido migrados, así como un resumen de este proceso (Tabla 15). Este artefacto se encuentra en Gestión de proyecto/ Implementación/. La información necesaria para la generación del mismo se puede encontrar en el subsistema de Migración, el cual posee, en la sección principal de cada uno de los proyectos existentes en la PCMCA, la información del avance de la migración de los sistemas operativos existentes en la entidad (ver Figura 10).

*Tabla 15: Migración de sistemas operativos*

Sistemas operativos
Cantidad de sistemas operativos
Cantidad migrados
Resumen

**10. Migracion\_estandares\_abiertos:** Este documento surge a partir del movimiento masivo de datos hacia la plataforma libre instalada, recopilando los más importantes durante este proceso. Para ello se tiene una tabla donde se recogen las extensiones privativas de los archivos existentes en la entidad, su alternativa libre y el porcentaje que representa cada uno del total (Tabla 16). Se creará una nueva fila por cada extensión existente en la empresa.

Este artefacto se encuentra en Gestión de proyecto/ Implementación/. La información necesaria para la generación del mismo se puede encontrar en el subsistema de Migración, el cual posee, en la sección principal de cada uno de los proyectos existentes en la PCMCA, la información del avance de la migración de los archivos existentes en la migración (ver Figura 10).

*Tabla 16: Migración de estándares abiertos*

Extensión privativa		Extensión libre		%
Extensión	Total	Extensión	Total	

## 1.5 Herramientas y Tecnologías

El subsistema de Migración de la PCMCA está desarrollado en el lenguaje PHP, con Symfony como



framework<sup>3</sup> de desarrollo y Bootstrap para darle estilo a la interfaz visual. Utiliza MongoDB como gestor de base de datos. Debido a que el componente será desarrollado dentro del subsistema de Migración de la PCMCA, se hará uso de las herramientas y tecnologías antes mencionadas, para lograr uniformidad en el subsistema.

- **MongoDB** es una base de datos ágil que permite a los esquemas cambiar rápidamente cuando las aplicaciones evolucionan, proporcionando siempre la funcionalidad que los desarrolladores esperan de las bases de datos tradicionales, tales como índices secundarios, un lenguaje completo de búsquedas y consistencia estricta. Ha sido creado para brindar escalabilidad, rendimiento y gran disponibilidad, escalando de una implantación de servidor único a grandes arquitecturas complejas de centros multidados. MongoDB brinda un elevado rendimiento, tanto para lectura como para escritura, potenciando la computación en memoria (in-memory). La replicación nativa de MongoDB y la tolerancia a fallos automática ofrece fiabilidad a nivel empresarial y flexibilidad operativa [4].
- **Symfony** es un framework diseñado para optimizar, gracias a sus características, el desarrollo de las aplicaciones web. Para empezar, separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación.

Symfony está desarrollado completamente con PHP y ha sido probado con éxito en sitios como Yahoo! Answers, Delicious, DailyMotion y muchos otros sitios web de primer nivel. Symfony es compatible con la mayoría de gestores de bases de datos, como MySQL, PostgreSQL, Oracle y SQL Server de Microsoft. Se puede ejecutar tanto en plataformas \*nix (Unix, Linux, etc.) como en plataformas Windows [5].

- **Bootstrap** es un framework o conjunto de herramientas de Código abierto para diseño de sitios y aplicaciones web. Contiene plantillas de diseño con tipografía, formularios, botones, cuadros, menús de navegación y otros elementos de diseño basado en HTML (HyperText Markup

---

3 *Framework: estructura de soporte definida, en la cual otro proyecto de software puede ser organizado*

Language, en español Lenguaje de Marcado para Hipertextos) y CSS<sup>4</sup> (Cascading Style Sheets, Hojas de Estilo en Cascada en español) [6].

- **PHP** es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML. Puede emplearse en todos los sistemas operativos principales, incluyendo Linux, muchas variantes de Unix (incluyendo HP-UX, Solaris y OpenBSD), Microsoft Windows, Mac OS X, RISC OS y probablemente otros más. PHP admite la mayoría de servidores web de hoy en día, incluyendo Apache, IIS, y muchos otros [7].
- **Odtphp** es una librería simple en el lenguaje PHP que permite cargar una plantilla de un documento de texto en formato ODT que contenga variables definidas en su contenido y cambiarlas por los valores necesarios.

## 1.6 Metodología de desarrollo de software

Para el desarrollo de este trabajo de diploma se hace uso de la metodología de desarrollo AUP-UCI, que es la definida para emplearse en los proyectos productivos de la UCI. Dicha definición se basa en una variación de la metodología “Proceso Unificado Ágil” (AUP por sus siglas en inglés) en unión con el modelo CMMI-DEV v 1.3. Esta variación de la metodología cuenta con 3 fases, las cuales se explican de forma breve a continuación:

1. **Inicio:** Durante el inicio del proyecto se llevan a cabo las actividades relacionadas con la planeación del proyecto. En esta fase se realiza un estudio inicial de la organización cliente que permite obtener información fundamental acerca del alcance del proyecto, realizar estimaciones de tiempo, esfuerzo y costo y decidir si se ejecuta o no el proyecto.
2. **Ejecución:** En esta fase se ejecutan las actividades requeridas para desarrollar el software, incluyendo el ajuste de los planes del proyecto considerando los requisitos y la arquitectura. Durante el desarrollo se modela el negocio, obtienen los requisitos, se elaboran la arquitectura y el diseño, se implementa y se libera el producto.
3. **Cierre:** En esta fase se analizan tanto los resultados del proyecto como su ejecución y se realizan

---

<sup>4</sup> *CSS: es un mecanismo simple que describe cómo se va a mostrar un documento en la pantalla, o cómo se va a imprimir, o incluso cómo va a ser pronunciada la información presente en ese documento a través de un dispositivo de lectura.*

las actividades formales de cierre del proyecto [8].

## **1.7 Conclusiones parciales**

El estudio realizado sobre la información existente en la PCMCA y la que se debe documentar en los artefactos del Expediente de Proyecto de Migración, permitió identificar los 10 artefactos que generará el componente y definir que esta información se encuentra específicamente en el subsistema de Migración. El estudio de este subsistema sentó las bases para establecer las herramientas a emplear: MongoDB, Symfony y Bootstrap, el lenguaje de programación PHP y la librería Odtphp para la creación de los artefactos. La metodología de desarrollo de software que se utilizará es AUP-UCI.

## Capítulo 2. Análisis y diseño

En este capítulo se define la propuesta de solución y se describe el funcionamiento del componente que se va a implementar, estableciendo los requisitos funcionales y no funcionales. Además se hace referencia a las funcionalidades del sistema mediante las Historias de Usuarios. Se define la arquitectura de desarrollo de software y los patrones de diseños utilizados, todo el proceso guiado por la metodología de desarrollo AUP-UCI.

### 2.1 Propuesta de la solución

Para darle solución al problema planteado se implementa, en el lenguaje PHP, un componente dentro del subsistema de Migración de la PCMCA que se encargará de generar los artefactos del Expediente de Proyecto de Migración con mayor rapidez y eficiencia. En la interfaz visual del componente se mostrará una lista con los 10 artefactos que se pueden generar desde la plataforma. Al seleccionar uno de los elementos de la lista se pasará a generar el artefacto correspondiente. Para lograr la creación de cada uno de ellos, el componente obtendrá la información requerida del subsistema de Migración, mediante el uso de los servicios ofrecidos por el mismo. Una vez obtenida esta información, se crearán los artefactos pasando los resultados a los distintos documentos usando la librería Odtphp.

### 2.2 Especificación de requisitos de software

En la tabla que a continuación se detalla, están registrados los 11 requisitos funcionales que tendrá la solución propuesta y 6 requisitos no funcionales.

Tabla 17: Especificación de requisitos

No.	Nombre	Descripción	Prioridad	Complejidad
<b>Requisitos Funcionales</b>				
1	Mostrar artefactos a generar	Se muestra una lista con los artefactos que se generan a partir del subsistema de Migración de la PCMCA	Alta	Baja

2	Generar artefacto Levantamiento_hardware	Genera el artefacto con la información de hardware del sistema	Alta	Alta
3	Generar artefacto Levantamiento_software	Genera el artefacto con la información de software del sistema	Alta	Alta
4	Generar artefacto Homologacion_hardware	Genera el artefacto con la información de los dispositivos homologados	Alta	Alta
5	Generar artefacto Levantamiento_dispositivos_externos	Genera el artefacto con la información de los dispositivos de entrada y salida	Alta	Alta
6	Generar artefacto Levantamiento_servicios_telematicos	Genera el artefacto con la información de los servicios telemáticos de la entidad	Media	Media
7	Generar artefacto Migracion_servicios_telematicos	Genera el artefacto con la información de los software que usan los servicios de red de la entidad.	Media	Media
8	Generar artefacto Migracion_servicios_telematicos(2)	Genera el artefacto con el porciento que representan los servicios de red que han sido migrados.	Baja	Baja
9	Generar artefacto Migracion_aplicaciones	Genera el artefacto con el porciento que representan las aplicaciones que han sido migradas.	Baja	Baja
10	Generar artefacto Migracion_sistema_operativo	Genera el artefacto con el porciento que representan los sistemas operativos que han sido migrados.	Baja	Baja
11	Generar artefacto Migracion_estandares_abiertos	Genera el artefacto con el porciento que representan los formatos libres.	Baja	Baja
<b>Requisitos no Funcionales</b>				
12	Usar librería Odtphp	Usar la librería Odtphp para generar los artefactos del expediente (Soporte).	Baja	Baja

13	Usar sobre ambiente Web.	El sistema podrá ser usado sobre ambiente Web por personas con conocimientos relacionados con los procesos de migración guiados por la Metodología Cubana de Migración a Código Abierto (Usabilidad).	Baja	Baja
14	Acceder desde diferentes navegadores	Se podrá acceder a la aplicación desde los navegadores web, Mozilla Firefox, Chrome y Opera (Confiabilidad).	Baja	Baja
15	Usar IDE PhpStorm	Utilizar como IDE PhpStorm en su versión 9.0 (Soporte).	Baja	Baja
16	Usar <i>framework</i> Symfony	<i>Framework</i> de desarrollo Symfony v1.4.18 (Soporte).	Baja	Baja
17	Usar lenguaje de programación PHP v5.	Los componentes desarrollados para la integración deberán ser implementados en lenguaje PHP v5 (Soporte).	Baja	Baja

## 2.3 Historias de Usuario

Para cada uno de los requisitos funcionales anteriores se ha definido una Historia de Usuario para describir su funcionamiento y demás características.

Tabla 18: Mostrar artefactos a generar

Historia de Usuario	
Número: HU_1	Nombre Historia de Usuario: Mostrar artefactos a generar
Modificación de Historia de Usuario Número: 1	
Usuario: Javier Junquera Falcón	Iteración Asignada: 2
Prioridad en Negocio: Alta	Puntos Estimados: 1

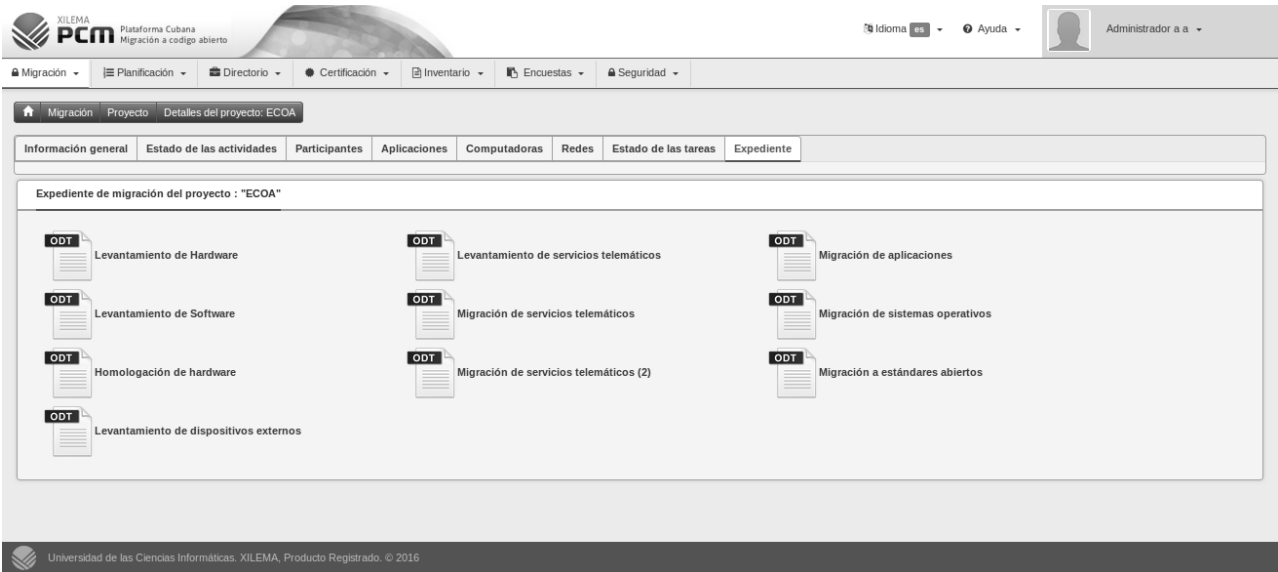
<b>Riesgo en Desarrollo:</b> Bajo	<b>Puntos Reales:</b>
<p><b>Descripción:</b> En las opciones del subsistema de Migración de la PCMCA se selecciona la opción Proyectos, que devuelve un listado con todos los proyectos existentes en la plataforma. Para poder continuar con este proceso debe existir al menos un proyecto registrado. Una vez se elige uno de estos proyectos, se selecciona la pestaña Expediente. Al dar clic en esta pestaña se mostrarán los 10 artefactos del expediente que se pueden generar.</p>	
<p><b>Observaciones:</b> Muestra la lista de los artefactos a generar</p>	
<p><b>Prototipo de interfaz:</b></p> 	

Tabla 19: Generar artefacto Levantamiento hardware

Historia de Usuario	
<b>Número:</b> HU_2	<b>Nombre Historia de Usuario:</b> Generar artefacto Levantamiento hardware
<b>Modificación de Historia de Usuario Número:</b> 1	
<b>Usuario:</b> Javier Junquera Falcón	<b>Iteración Asignada:</b> 1
<b>Prioridad en Negocio:</b> Alta	<b>Puntos Estimados:</b> 1

<b>Riesgo en Desarrollo:</b> Alto	<b>Puntos Reales:</b>
<p><b>Descripción:</b> Al seleccionar el elemento Levantamiento de Hardware de la lista de artefactos a generar, comienza el proceso para generar este artefacto. Usando el servicio listarPCs(\$_SESSION['apm']['idproyecto']) se obtiene un listado de todas las computadoras registradas en el proyecto actual. Una vez se tiene esta lista, para cada uno de los elementos de la misma, se obtiene su identificador y el responsable de cada PC y se usa el servicio ListarHardwarePCByIdsPC() para obtener un listado con todos los dispositivos de hardware que se encuentren en cada PC. Con este listado se pueden separar los diferentes dispositivos según su tipo y de cada uno de ellos se obtiene la información necesaria para generar el artefacto. Usando la librería <i>Odtphp</i> se carga la plantilla del artefacto a generar y se pasa la información obtenida al mismo, luego se genera en formato Odt.</p>	
<p><b>Observaciones:</b> Genera el artefacto: Levantamiento hardware</p>	
<p><b>Prototipo de interfaz:</b> --</p>	

Tabla 20: Generar artefacto Levantamiento software

Historia de Usuario	
<b>Número:</b> HU_3	<b>Nombre Historia de Usuario:</b> Generar artefacto Levantamiento software
<b>Modificación de Historia de Usuario Número:</b> 1	
<b>Usuario:</b> Javier Junquera Falcón	<b>Iteración Asignada:</b> 1
<b>Prioridad en Negocio:</b> Alta	<b>Puntos Estimados:</b> 1
<b>Riesgo en Desarrollo:</b> Alto	<b>Puntos Reales:</b>
<p><b>Descripción:</b> Al seleccionar el elemento Levantamiento de Software de la lista de artefactos a generar, comienza el proceso para generar este artefacto. Se obtienen todas las computadoras del proyecto usando el servicio listarPCs(\$_SESSION['apm']['idproyecto']). Para cada uno de los elementos de esta lista de PCs se usa el servicio ListarAplicacionComputadora() para obtener una lista con todas las aplicaciones que pertenecen a las mismas. Una vez obtenida la lista se busca si estas son aplicaciones propias de la entidad y si son aplicaciones críticas para el funcionamiento de la misma. De cada aplicación se obtiene el nombre y la versión. Usando la librería <i>Odtphp</i> se carga la plantilla del artefacto a generar y se pasa la información obtenida al mismo, luego se genera en formato Odt.</p>	
<p><b>Observaciones:</b> Genera el artefacto Levantamiento software</p>	
<p><b>Prototipo de interfaz:</b></p>	



Tabla 21: Generar artefacto Homologacion\_hardware

Historia de Usuario	
Número: HU_4	Nombre Historia de Usuario: Generar artefacto Homologacion_hardware
Modificación de Historia de Usuario Número: 1	
Usuario: Javier Junquera Falcón	Iteración Asignada: 1
Prioridad en Negocio: Alta	Puntos Estimados: 1
Riesgo en Desarrollo: Alto	Puntos Reales:
<p><b>Descripción:</b> Al seleccionar el elemento Homologación de Hardware de la lista de artefactos a generar, comienza el proceso para generar este artefacto. Usando el servicio listarPCs(\$_SESSION['apm']['idproyecto']) se obtiene un listado de todas las computadoras que pertenecen a una entidad determinada. Una vez se tiene esta lista, para cada uno de los elementos de la misma, se usa el servicio ListarDispositivosHomologadosByPCs() para obtener un listado con todos los dispositivos de hardware que se encuentren en dicha PC que han sido homologados. Con este listado se pueden separar los diferentes dispositivos según si son externos o internos. Usando la librería <i>Odtphp</i> se carga la plantilla del artefacto a generar y se pasa la información obtenida al mismo, luego se genera en formato Odt.</p>	
<b>Observaciones:</b> Genera el artefacto: Homologacion_hardware	
<b>Prototipo de interfaz:</b>	

Tabla 22: Generar artefacto Levantamiento\_dispositivos\_externos

Historia de Usuario	
Número: HU_5	Nombre Historia de Usuario: Generar artefacto Levantamiento_dispositivos_externos
Modificación de Historia de Usuario Número: 1	
Usuario: Javier Junquera Falcón	Iteración Asignada: 1
Prioridad en Negocio: Alta	Puntos Estimados: 1
Riesgo en Desarrollo: Alto	Puntos Reales:
<p><b>Descripción:</b> Al seleccionar el elemento Levantamiento de Dispositivos Externos de la lista de artefactos a generar,</p>	

comienza el proceso para generar este artefacto. Usando el servicio listarPCs(\$\_SESSION['apm']['idproyecto']) se obtiene un listado de todas las computadoras que pertenecen a una entidad determinada. Una vez se tiene esta lista, para cada uno de los elementos de la misma, se usa el servicio ListarHardwarePCByIdsPC() para obtener un listado con todos los dispositivos de hardware que se encuentren en dicha PC. Con este listado se pueden separar los diferentes dispositivos según si son de entrada, de salida o de entrada/salida. Usando la librería *Odtphp* se carga la plantilla del artefacto a generar y se pasa la información obtenida al mismo, luego se genera en formato Odt.

**Observaciones:** Genera el artefacto: Levantamiento\_dispositivos\_externos

**Prototipo de interfaz:**

Tabla 23: Generar artefacto Levantamiento\_servicios\_telematicos

Historia de Usuario	
<b>Número:</b> HU_7	<b>Nombre Historia de Usuario:</b> Generar artefacto Levantamiento_servicios_telematicos
<b>Modificación de Historia de Usuario Número:</b> 1	
<b>Usuario:</b> Javier Junquera Falcón	<b>Iteración Asignada:</b> 1
<b>Prioridad en Negocio:</b> Medio	<b>Puntos Estimados:</b> 1
<b>Riesgo en Desarrollo:</b> Medio	<b>Puntos Reales:</b>
<b>Descripción:</b> Al seleccionar el elemento Levantamiento de Servicios Telemáticos de la lista de artefactos a generar, comienza el proceso para generar este artefacto. Se usa el servicio listarServicioRed() para obtener todos los servicios telemáticos de un proyecto determinado. Una vez se tienen estos servicios, se buscan el sistema operativo sobre el que funcionan, la cantidad de usuarios que permite que lo usen de forma concurrente y la cantidad total de usuarios que lo utilizan. Usando la librería <i>Odtphp</i> se carga la plantilla del artefacto a generar y se pasa la información obtenida al mismo, luego se genera en formato Odt.	
<b>Observaciones:</b> Genera el artefacto: Levantamiento_servicios_telematicos	
<b>Prototipo de interfaz:</b>	

Tabla 24: Generar artefacto Migracion\_servicios\_telematicos

Historia de Usuario	
<b>Número:</b> HU_8	<b>Nombre Historia de Usuario:</b> Generar artefacto

	Migracion_servicios_telematicos
<b>Modificación de Historia de Usuario Número: 1</b>	
<b>Usuario:</b> Javier Junquera Falcón	<b>Iteración Asignada:</b> 1
<b>Prioridad en Negocio:</b> Media	<b>Puntos Estimados:</b> 1
<b>Riesgo en Desarrollo:</b> Medio	<b>Puntos Reales:</b>
<b>Descripción:</b> Al seleccionar el elemento Migración de servicios telemáticos de la lista de artefactos a generar, comienza el proceso para generar este artefacto. Se usa el servicio listarServicioRed() para obtener todos los servicios telemáticos de un proyecto determinado y de cada uno de estos se adquiere el software que utiliza cada uno de estos servicios y el sistema operativo actual del mismo. Usando la librería <i>Odtphp</i> se carga la plantilla del artefacto a generar y se pasa la información obtenida al mismo, luego se genera en formato Odt.	
<b>Observaciones:</b> Generar el artefacto: Migracion_servicios_telematicos	
<b>Prototipo de interfaz:</b>	

Tabla 25: Generar artefacto Migracion\_servicios\_telematicos(2)

Historia de Usuario	
<b>Número:</b> HU_9	<b>Nombre Historia de Usuario:</b> Generar artefacto Migracion_servicios_telematicos(2)
<b>Modificación de Historia de Usuario Número: 1</b>	
<b>Usuario:</b> Javier Junquera Falcón	<b>Iteración Asignada:</b> 1
<b>Prioridad en Negocio:</b> Baja	<b>Puntos Estimados:</b> 1
<b>Riesgo en Desarrollo:</b> Bajo	<b>Puntos Reales:</b>
<b>Descripción:</b> Al seleccionar el elemento Migración de Servicios Telemáticos(2) de la lista de artefactos a generar, comienza el proceso para generar este artefacto. Se utiliza el servicio listarProyectoByIdsPCS() para obtener los datos del proyecto requerido. De este proyecto se obtienen el total de servicios telemáticos, la cantidad que han sido migrados y se calcula el porcentaje que representa esta última del total. Usando la librería <i>Odtphp</i> se carga la plantilla del artefacto a generar y se pasa la información obtenida al mismo, luego se genera en formato Odt.	
<b>Observaciones:</b> Genera el artefacto: Migracion_servicios_telematicos(2)	
<b>Prototipo de interfaz:</b>	

Tabla 26: Generar artefacto Migracion\_Aplicaciones

Historia de Usuario	
<b>Número:</b> HU_10	<b>Nombre Historia de Usuario:</b> Generar artefacto Migracion_Aplicaciones
<b>Modificación de Historia de Usuario Número: 1</b>	
<b>Usuario:</b> Javier Junquera Falcón	<b>Iteración Asignada:</b> 1
<b>Prioridad en Negocio:</b> Baja	<b>Puntos Estimados:</b> 1
<b>Riesgo en Desarrollo:</b> Bajo	<b>Puntos Reales:</b>
<b>Descripción:</b> Al seleccionar el elemento Migración de Aplicaciones de la lista de artefactos a generar, comienza el proceso para generar este artefacto. Se utiliza el servicio listarProyectoByIdsPCS() para obtener los datos del proyecto requerido. De este proyecto se obtienen el total de aplicaciones, la cantidad que han sido migradas y se calcula el porcentaje que representa esta última del total. Usando la librería <i>Odtphp</i> se carga la plantilla del artefacto a generar y se pasa la información obtenida al mismo, luego se genera en formato Odt.	
<b>Observaciones:</b> Genera el artefacto: Migracion_Aplicaciones	
<b>Prototipo de interfaz:</b>	

Tabla 27: Generar artefacto Migracion\_Sistema\_Operativo

Historia de Usuario	
<b>Número:</b> HU_11	<b>Nombre Historia de Usuario:</b> Generar artefacto Migracion_Sistema_Operativo
<b>Modificación de Historia de Usuario Número: 1</b>	
<b>Usuario:</b> Javier Junquera Falcón	<b>Iteración Asignada:</b> 1
<b>Prioridad en Negocio:</b> Baja	<b>Puntos Estimados:</b> 1
<b>Riesgo en Desarrollo:</b> Bajo	<b>Puntos Reales:</b>
<b>Descripción:</b> Al seleccionar el elemento Migración de sistemas operativos de la lista de artefactos a generar, comienza el proceso para generar este artefacto. De este proyecto se obtienen el total de sistemas operativos, la cantidad que han sido migrados y se calcula el porcentaje que representa esta última del total. Usando la librería <i>Odtphp</i> se carga la plantilla del artefacto a generar y se pasa la información obtenida al mismo, luego se genera en	

formato Odt.
<b>Observaciones:</b> Generar el artefacto: Migracion_Sistema_Operativo
<b>Prototipo de interfaz:</b>

Tabla 28: Generar artefacto Migracion\_estandares\_abiertos

Historia de Usuario	
<b>Número:</b> HU_12	<b>Nombre Historia de Usuario:</b> Generar artefacto Migracion_estandares_abiertos
<b>Modificación de Historia de Usuario Número:</b> 1	
<b>Usuario:</b> Javier Junquera Falcón	<b>Iteración Asignada:</b> 1
<b>Prioridad en Negocio:</b> Baja	<b>Puntos Estimados:</b> 1
<b>Riesgo en Desarrollo:</b> Bajo	<b>Puntos Reales:</b>
<b>Descripción:</b> Al seleccionar el elemento Migración a estándares abiertos de la lista de artefactos a generar, comienza el proceso para generar este artefacto. De este proyecto se obtienen el total de aplicaciones, la cantidad que han sido migradas y se calcula el porcentaje que representa esta última del total. Usando la librería <i>Odtphp</i> se carga la plantilla del artefacto a generar y se pasa la información obtenida al mismo, luego se genera en formato Odt.	
<b>Observaciones:</b> Genera el artefacto: Migracion_estandares_abiertos	
<b>Prototipo de interfaz:</b>	

## 2.4 Arquitectura de software

La arquitectura es la estructura jerárquica de los componentes del programa, la manera en que los componentes interactúan y la estructura de datos que van a utilizar los componentes [9]. La PCMCA utiliza dos estilos arquitectónicos fundamentalmente: SOA (*Service Oriented Architecture*, en español Arquitectura Orientada a Servicios) y Presentación Desacoplada [10]. Sin embargo, cada subsistema utiliza de manera interna una arquitectura n-Capas, incluyendo el subsistema de Migración. El estilo arquitectónico n-Capas se basa en una distribución jerárquica de los roles y responsabilidades para proporcionar una división efectiva de los problemas a resolver. Los roles indican el tipo y la forma de interacción con otras capas, mientras que las responsabilidades indican la funcionalidad que implementan

[11]. En la Figura 11 se puede observar el diagrama de componentes correspondiente al subsistema de Migración, donde se muestra la integración de este subsistema (paquete MAGPM) a la plataforma (paquete núcleo SINPM).

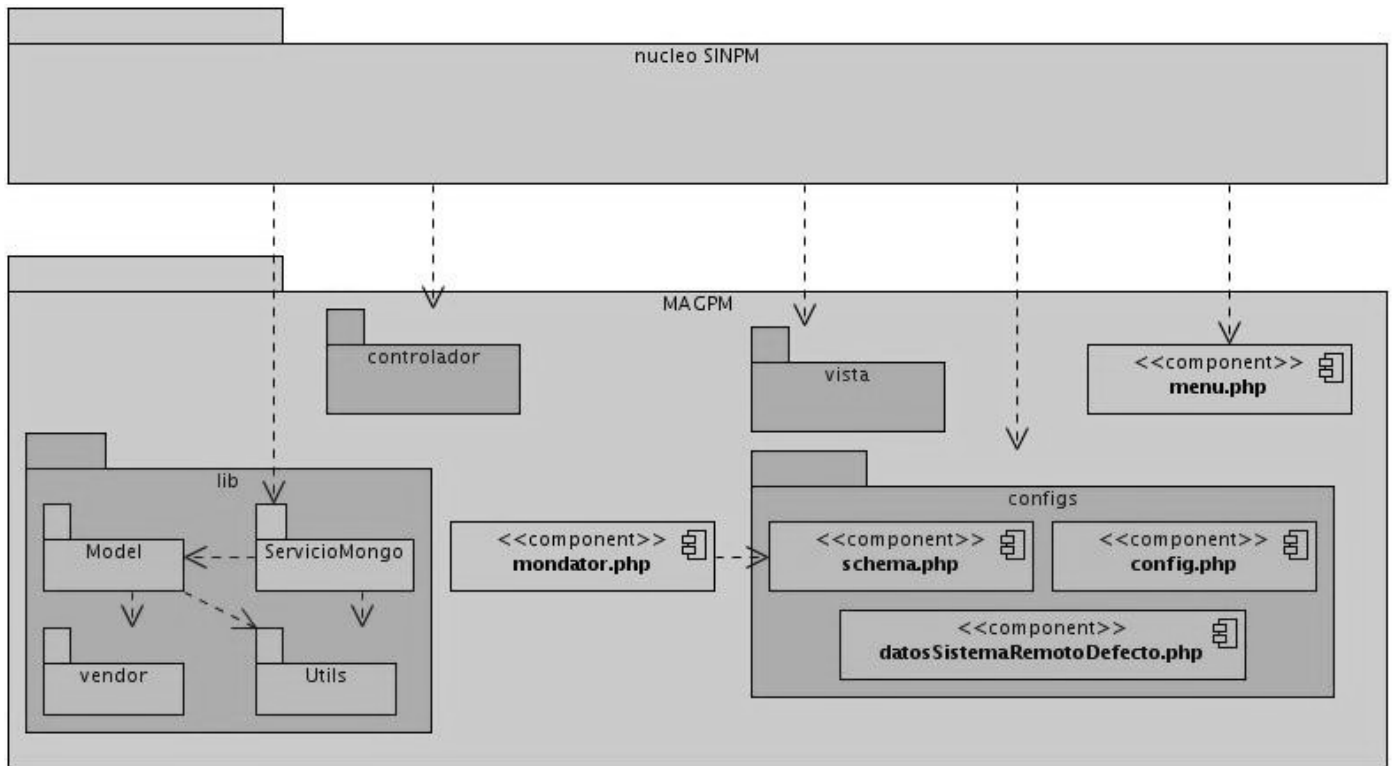


Figura 11: Diagrama de componentes del subsistema de Migración [3]

El componente a desarrollar, al ser parte del subsistema de Migración, usará la arquitectura n-Capas, usando el estilo arquitectónico 2-Capas. Existen dos paquetes, uno para la capa Controlador, en la que existirá la clase Expediente, que contendrá los métodos para generar los artefactos definidos. El segundo es para la capa Vista, donde se implementará la interfaz visual del componente (Figura 12).

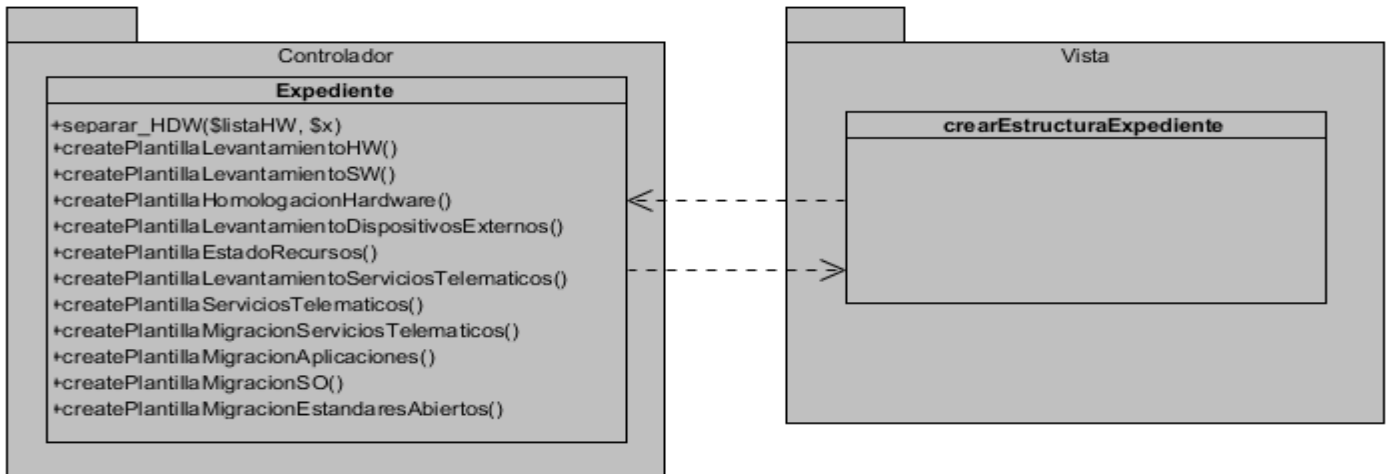


Figura 12: Arquitectura del componente para la generación de los artefactos del expediente de proyecto

## 2.5 Patrones de diseño

Los Patrones de Software de Asignación de Responsabilidades Generales (GRASP, por sus siglas en inglés) son patrones que ayudan en el desarrollo de un software al facilitar el uso de la arquitectura escogida, más que patrones de diseño, se consideran buenas prácticas en el proceso de desarrollo de software [12]. En el desarrollo del componente se usan dos de estos patrones:

- **Creador:** Es el encargado de guiar la asignación de responsabilidades relacionadas con la creación de objetos [12].

En los siguientes ejemplos de código se ejemplifica cómo se crea un nuevo objeto de la clase CoreCapaServicio y un objeto de la clase CoreServicio respectivamente.

Ejemplos: `$servicio = new nucleo\CoreCapaServicio('ServicioMongo\Hardware');`

`$servicio = new nucleo\CoreServicio('pcs');`

- **Alta Cohesión:** Caracteriza a las clases con gran cantidad de funcionalidades, se dividieron en otras clases siguiendo el propósito de distribuir de forma equitativa el peso de la complejidad, manteniendo además, la coherencia entre ellas [12].

En el ejemplo siguiente se muestran dos fragmentos de código en los cuales se observa cómo la clase CoreAccion se divide en otra clase llamada Expediente debido al alto número de funcionalidades que posee.

*Ejemplo:* `class` CoreAccion

```
class Expediente extends nucleo\CoreAccion
```

## 2.6 Conclusiones parciales

A partir del análisis realizado al subsistema de Migración y una vez determinados los 10 artefactos del Expediente de Proyecto de Migración a generar, se definieron los requisitos funcionales y no funcionales del componente. Con el empleo de la metodología AUP-UCI se especificaron las HU. Partiendo de la arquitectura del subsistema de Migración, se propuso la arquitectura a emplear en el desarrollo del componente.



## Capítulo 3. Implementación y pruebas

Durante todo proceso de desarrollo de software es necesario comprobar la calidad del mismo. Primeramente se realiza la implementación para luego pasar a realizar un conjunto de pruebas para encontrar todos los errores que aún puedan existir en el software desarrollado. En el siguiente capítulo se define el estándar de codificación usado en la implementación y los resultados de varias pruebas que se realizaron para asegurar la calidad del producto.

### 3.1 Implementación

Durante la implementación del componente se siguieron varias reglas definidas por los estándares de codificación del lenguaje escogido, definidos por la comunidad PHP, para lograr la creación de un software cuyo código sea fácil de entender. También se detectó un problema durante la implementación del requisito funcional número 2. Ambos casos se describen en los epígrafes siguientes.

#### 3.1.1 Métodos implementados

- `public function` separar\_HW()

Este método recibe un listado de dispositivos de hardware y separa cada uno de estos dispositivos por tipo.

```
foreach($listaHW[$x]['listHardwarePC'] as $objethardware)
{
    $tipohw = $objethardware->getDispositivo()->getTipo();
    if($tipohw=='disco')
        $listadisco[] = $objethardware->getDispositivo();
    else if($tipohw=='procesador')
        $listaproc[] = $objethardware->getDispositivo();
    else if($tipohw=='red')
        $listared[] = $objethardware->getDispositivo();
}
```

```

else if($tipohw=='audio')
    $listaaudio[] = $objethardware->getDispositivo();
else if($tipohw=='video')
    $listavideo[] = $objethardware->getDispositivo();
else if($tipohw=='monitor')
    $listamonitor[] = $objethardware->getDispositivo();
else if($tipohw=='RAM')
    $listaram[] = $objethardware->getDispositivo();
else if($tipohw=='bios')
    $listabios[] = $objethardware->getDispositivo();
}

```

De cada uno de estos dispositivos se obtienen los datos requeridos por los artefactos Levantamiento\_hardware, Homologacion\_hardware y Levantamiento\_dispositivos\_externos.

- **public function** createPlantillaLevantamientoHW()

```
$listapc = $servicio->listarPCs($_SESSION['apm']['idProyecto']);
```

Usando el servicio anterior se obtiene un listado de todas las computadoras registradas en el proyecto actual. Una vez se tiene esta lista, para cada uno de los elementos de la misma, se obtiene su identificador y el responsable de cada PC.

```

for($i= 0; $i<= count($listapc); $i++)
{
    $array_id[]= $listapc['listaPC'][$i]->getID();
    $nombreResponsable[]= $listapc['listaPC'][$i]->getResponsable()
    ->getNombreApellido();
}

```

Se obtiene un listado con todos los dispositivos de hardware que se encuentren en cada PC.

```
$result[] = $servicio->listarHardwarePCByIdsPC($res);
```

Se hace la llamada al método `separar_HW()` que recibe el listado anterior para separar los diferentes dispositivos según su tipo y de cada uno de ellos se obtiene la información necesaria para generar el artefacto. Usando la librería *Odtphp* se carga la plantilla del artefacto a generar y se pasa la información obtenida al mismo, luego se genera en formato Odt.

- **public function** `createPlantillaLevantamientoSW()`

Se obtienen todas las computadoras del proyecto usando el servicio `listarPCs($_SESSION['apm']['idproyecto'])`. Para cada uno de los elementos de esta lista de PCs se obtiene una lista con todas las aplicaciones que pertenecen a las mismas.

```
$lista_apps[] = $servicio->listarAplicacionComputadora(NULL, NULL, $aplicaciones = array(), $array_id ,NULL);
```

Una vez obtenida la lista se busca si estas son aplicaciones propias de la entidad y si son aplicaciones críticas para el funcionamiento de la misma.

```
foreach($listaAplicaciones as $element)
{
    $resp = $element->getPropio() ? 'Si' : 'No';
    if($resp == 'Si')
        $listaPropias[] = $element;
}
```

```
if($aplicacion->esUtil())
    $listaUtiles[] = $aplicacion;
```

- De cada aplicación se obtiene el nombre y la versión.

```
for($i = 0; $i < count($listaAplicaciones); $i++)
{
    $apps->nombre($listaAplicaciones[$i]->getNombre());
    $apps->version($listaAplicaciones[$i]->getVersion());
}
```

```
    $apps->merge();  
}
```

Usando la librería *Odtphp* se carga la plantilla del artefacto a generar y se pasa la información obtenida al mismo, luego se genera en formato Odt.

- **public function** createPlantillaHomologacionHardware()

Al seleccionar el elemento Homologación de Hardware de la lista de artefactos a generar, se hace la llamada a este método. Usando el servicio listarPCs(\$\_SESSION['apm']['idproyecto']) se obtiene un listado de todas las computadoras que pertenecen a una entidad determinada. Una vez se tiene esta lista, para cada uno de los elementos de la misma, se obtiene un listado con todos los dispositivos de hardware que se encuentren en dicha PC que han sido homologados.

```
$result[] = $servicio->listarDispositivosHomologadosByPCs($res);  
  
$tipohw = $objethardware->getDispositivo()->getTipo();  
if($tipohw=='disco' || $tipohw=='video' || $tipohw=='monitor')  
    $listaexternos = $objethardware->getDispositivo();  
else if($tipohw=='procesador' || $tipohw=='red' || $tipohw=='audio' ||  
    $tipohw=='RAM' || $tipohw=='bios')  
    $listainternos = $objethardware->getDispositivo();
```

Se hace la llamada al método separar\_HW() que recibe el listado anterior para separar los diferentes dispositivos según su tipo y de cada uno de ellos se obtiene la información necesaria para generar el artefacto. Con este listado se pueden separar los diferentes dispositivos según si son externos o internos. Usando la librería *Odtphp* se carga la plantilla del artefacto a generar y se pasa la información obtenida al mismo, luego se genera en formato Odt.

- **public function** createPlantillaLevantamientoDispositivosExternos()

Al seleccionar el elemento Levantamiento de Dispositivos Externos de la lista de artefactos a generar, se hace la llamada a este método. Usando el servicio listarPCs(\$\_SESSION['apm']

['idproyecto']) se obtiene un listado de todas las computadoras que pertenecen a una entidad determinada. Una vez se tiene esta lista, para cada uno de los elementos de la misma, se usa el servicio `ListarHardwarePCByIdsPC()` para obtener un listado con todos los dispositivos de hardware que se encuentren en dicha PC. Se hace la llamada al método `separar_HW()` que recibe el listado anterior para separar los diferentes dispositivos según su tipo y de cada uno de ellos se obtiene la información necesaria para generar el artefacto. Luego se separan los diferentes dispositivos según si son de entrada, de salida o de entrada/salida. Usando la librería *OdtpHP* se carga la plantilla del artefacto a generar y se pasa la información obtenida al mismo, luego se genera en formato Odt.

- `public function createPlantillaLevantamientoServiciosTelematicos()`

Al seleccionar el elemento Levantamiento de Servicios Telemáticos de la lista de artefactos a generar, se llama este método. Se obtienen todos los servicios telemáticos de un proyecto determinado.

```
$listaST = $servicio->listarServicioRed($_SESSION['apm']['idProyecto']);
```

Una vez se tienen estos servicios, se buscan el sistema operativo sobre el que funcionan, la cantidad de usuarios que permite que lo usen de forma concurrente y la cantidad total de usuarios que lo utilizan.

```
foreach($listaST['listaServicioRed'] as $servred)
```

```
{
```

```
    $nombreserv[] = $servred->getNombre();
```

```
    $pcserv[] = $servred->getComputadoras();
```

```
    $appserv[] = $servred->getAplicaciones();
```

```
    $observ[] = $servred->getObservaciones();
```

```
    $relacion = $servred->getRelacion();
```

```
    $S0red = $servred->getComputadoras()->getComputadora()
```

```
    ->getNombreSistemaOperativo();
```

```
}
```

Usando la librería *Odtphp* se carga la plantilla del artefacto a generar y se pasa la información obtenida al mismo, luego se genera en formato Odt.

- **public function** createPlantillaServiciosTelematicos()

Al seleccionar el elemento Migración de servicios telemáticos de la lista de artefactos a generar, se llama este método. Se usa el servicio listarServicioRed() para obtener todos los servicios telemáticos de un proyecto determinado y de cada uno de estos se adquiere el software que utiliza cada uno de estos servicios y el sistema operativo actual del mismo.

```
foreach($listaST['listaServicioRed'] as $servred)
{
    $nombreserv[] = $servred->getNombre();
    $solucion = $servred->getSolucion()->getServicioRed();
    $SOfinal = $solucion->getComputadoras()->getComputadora()
    ->getNombreSistemaOperativo();
    $softFinal = $solucion->getAplicaciones();
    $appserv[] = $servred->getAplicaciones();
    $observ[] = $servred->getObservaciones();
    $SOfinal = $servred->getComputadoras()->getComputadora()
    ->getNombreSistemaOperativo();
}
```

Usando la librería *Odtphp* se carga la plantilla del artefacto a generar y se pasa la información obtenida al mismo, luego se genera en formato Odt.

- **public function** createPlantillaMigracionServiciosTelematicos()

Al seleccionar el elemento Migración de Servicios Telemáticos(2) de la lista de artefactos a generar, se ejecuta este método. Se utiliza el servicio listarProyectoByIdsPCS() para obtener los datos del proyecto requerido.

```
$proyectosPCS = $servicioProyecto
->listarProyectoByIdsPCS(array($_SESSION['apm']['idProyectoPCS']));
```

De este proyecto se obtienen el total de servicios telemáticos, la cantidad que han sido migrados.

```
$cant_servicios = $this->proyectoLocal->getRedEntidad()
->getServiciosRed()->count();
```

```
$servicios_migrados = $this->proyectoLocal
->getSolucionServiciosRedCount();
```

Usando la librería *Odtphp* se carga la plantilla del artefacto a generar y se pasa la información obtenida al mismo, luego se genera en formato Odt.

- **public function** createPlantillaMigracionAplicaciones()

Al seleccionar el elemento Migración de Aplicaciones de la lista de artefactos a generar, se hace la llamada a este método. Se utiliza el servicio listarProyectoByIdsPCS() para obtener los datos del proyecto requerido. De este proyecto se obtienen el total de aplicaciones, la cantidad que han sido migradas y se calcula el porcentaje que representa esta última del total. `$cant_aplicaciones = $this->proyectoLocal->getAplicaciones()->count();`

```
$aplicaciones_migradas = $this->proyectoLocal->getAplicacionesMigradasCount();
if($cant_aplicaciones!=0)
$porcentaje_aplicaciones = $aplicaciones_migradas/$cant_aplicaciones *100;
```

Usando la librería *Odtphp* se carga la plantilla del artefacto a generar y se pasa la información obtenida al mismo, luego se genera en formato Odt.

- `public function createPlantillaMigracionSO()`

Al seleccionar el elemento Migración de sistemas operativos de la lista de artefactos a generar, se ejecuta este método. De este proyecto se obtienen el total de sistemas operativos, la cantidad que han sido migrados.

```
$cantSO = $this->proyectoLocal->getSOsCount();
```

```
$SOmigrados = $this->proyectoLocal->getMigracionesSO()->count();
```

Usando la librería *Odtphp* se carga la plantilla del artefacto a generar y se pasa la información obtenida al mismo, luego se genera en formato Odt.

- `public function createPlantillaMigracionEstandaresAbiertos()`

Al seleccionar el elemento Migración a estándares abiertos de la lista de artefactos a generar, comienza el proceso para generar este artefacto. De este proyecto se obtienen el total de aplicaciones, la cantidad que han sido migradas y se calcula el porcentaje que representa esta última del total. Usando la librería *Odtphp* se carga la plantilla del artefacto a generar y se pasa la información obtenida al mismo, luego se genera en formato Odt.

### 3.1.2 Estándar de codificación

La comunidad detrás de PHP es muy diversa, por lo que está compuesta de innumerables librerías, *frameworks* y componentes. Así que es común que combinar varios proyectos PHP diferentes en una misma aplicación. Por eso es importante que los diferentes proyectos utilicen, en la medida de lo posible, el mismo estilo para escribir su código [13]. El *Framework Interop Group* (antes conocido como el *PHP Standards Group*) ha propuesto y aprobado una serie de recomendaciones de estilo, conocidas como *PHP Standard Recommendations* (PSR), las cuales son PSR-1, PSR-2, PSR-3 y PSR-4. Durante la implementación del componente se usaron los estándares PSR-1 y PSR-2 los cuales serán descritos a continuación.

#### PSR-1 - Estándar básico de estilos de código

Orientado al contenido de los ficheros PHP y a los nombres de las clases y métodos. Su objetivo es



garantizar un alto nivel técnico de interoperabilidad entre el código PHP.

- Deben usarse únicamente las etiquetas `<?php` y `<?=>`.
- Los Namespaces y las clases deben cumplir el estándar PSR-4.
- Los nombres de las clases deben utilizar la notación *StudlyCaps*<sup>5</sup>.

*Ejemplos:* `class Expediente extends nucleo\CoreAccion`

- Los nombres de los métodos deben declararse en notación *camelCase*<sup>6</sup>.

*Ejemplos:* `public function createPlantillaLevantamientoHW(){...}`

- Para los nombres de las propiedades no se define una recomendación concreta. A excepción de que la convención que se elija se mantenga para todo el proyecto, clase o método [14].

## PSR-2 - Guía de estilo de código

Orientado a dar formato al código PHP. Su objetivo es reducir la dificultad cuando se lee código de diferentes autores.

### Código

- El código debe seguir el estándar PSR-1.
- No hay un límite estricto en la longitud de las líneas. El límite debe estar en 120 caracteres; las líneas deberían tener preferiblemente 80 caracteres o menos.
  - ◆ No deben haber espacios en blanco al final de las líneas que no estén vacías.
  - ◆ Pueden añadirse líneas en blanco para mejorar la lectura del código y para indicar bloques de código que estén relacionados.
  - ◆ No debe haber más de una sentencia por línea.

---

5 *StudlyCaps es una notación en la que se alternan mayúsculas y minúsculas por algún patrón concreto.*

6 *En camelCase se escribe la primera letra de cada palabra con mayúsculas (Habitualmente la primera letra de todas suele ir en minúsculas).*

## Clases

- Las llaves de apertura de las clases deben ir en la línea siguiente, y las llaves de cierre deben ir en la línea siguiente al cuerpo de la clase.
- Las llaves de apertura de los métodos deben ir en la línea siguiente, y las llaves de cierre deben ir en la línea siguiente al cuerpo del método.

Ejemplo: `class Expediente extends nucleo\CoreAccion`

```
{
    public function createPlantillaLevantamientoHW()
    {
        $servicio = new nucleo\CoreCapaServicio('ServicioMongo\Hardware');
    }
}
```

- Las palabras clave *extends* e *implements* deben declararse en la misma línea del nombre de la clase [15].

Ejemplo: `class Expediente extends nucleo\CoreAccion`

### 3.1.3 Cambios en el requisito funcional 2

Durante la implementación del componente se detectó que la librería Odtphp no posee las funcionalidades necesarias para repetir una tabla completa en un documento odt. Debido a esto, es imposible generar el artefacto Levantamiento\_hardware en su totalidad. Por tanto se decide modificar el artefacto ya que la segunda tabla del mismo debe ser repetida por cada computadora existente en la entidad. Se mantendrán los mismos datos en la segunda parte del artefacto, solo que la información no estará contenida en una tabla. Así será posible generar el artefacto con la misma librería que se usa para los demás y no se perderá información en el artefacto. La segunda sección del artefacto quedará con la estructura siguiente:

**Nombre del Responsable:**

**Hardware de Almacenamiento**

**Modelo** \_\_\_\_\_ **Capacidad**

**Bios**

**Fabricante** \_\_\_\_\_ **Modelo**

**Memoria RAM**

**Capacidad** \_\_\_\_\_ **Velocidad** \_\_\_\_\_ **Tipo**

**Monitor**

**Fabricante** \_\_\_\_\_ **Modelo** \_\_\_\_\_ **Tipo**

**Procesador**

**Tipo** \_\_\_\_\_ **Velocidad**

**Red**

**Modelo** \_\_\_\_\_ **Velocidad**

**Tarjeta Sonido**

**Fabricante**

**Tarjeta de Video**

**Modelo** \_\_\_\_\_ **Memoria**

**3.2 Definición de las pruebas**

Las pruebas del software son un elemento crítico para la garantía de calidad del software y representan

una revisión final de las especificaciones, del diseño y de la codificación. El objetivo de las pruebas, expresado de forma sencilla, es encontrar el mayor número posible de errores con una cantidad razonable de esfuerzo, aplicado sobre un lapso de tiempo realista [9]. Para conseguir este objetivo, se planifica y se ejecutan una serie de pruebas definidas por la metodología AUP-UCI; las cuales son las pruebas internas y pruebas de aceptación.

### **3.2.1 Pruebas internas**

Para la realización de las pruebas internas, las cuales permiten verificar el resultado de la implementación, se decide usar el método de caja blanca, y más específico la técnica del camino básico. La prueba de caja blanca, denominada a veces prueba de caja de cristal, es un método de diseño de casos de prueba que usa la estructura de control del diseño procedimental para obtener los casos de prueba. Mediante los métodos de prueba de caja blanca, el ingeniero del software puede obtener casos de prueba que:

1. Garanticen que se ejercita por lo menos una vez todos los caminos independientes de cada módulo.
2. Ejerciten todas las decisiones lógicas en sus vertientes verdadera y falsa.
3. Ejecuten todos los bucles en sus límites y con sus límites operacionales.
4. Ejerciten las estructuras internas de datos para asegurar su validez [9].

El método del camino básico permite al diseñador de casos de prueba obtener una medida de la complejidad lógica de un diseño procedimental y usar esa medida como guía para la definición de un conjunto básico de caminos de ejecución. Los casos de prueba obtenidos del conjunto básico garantizan que durante la prueba se ejecuta por lo menos una vez cada sentencia del programa.

Para comenzar a realizar la técnica del camino básico se divide el código y se le asigna una numeración a cada bloque para luego crear el Grafo de Flujo de dicha funcionalidad (Figura 8). A continuación se muestra el código fuente del método `createPlantillaLevantamientoHW` del componente desarrollado.

Tabla 29: Código de la funcionalidad createPlantillaLevantamientoHW

	<code>public function createPlantillaLevantamientoHW() {</code>
1	<code>  \$servicio = new nucleo\CoreCapaServicio('ServicioMongo\Hardware');   \$listapc = \$servicio-&gt;listarPCs(\$_SESSION['apm']['idProyecto']);</code>
2	<code>    for(\$i= 0; \$i&lt;= count(\$listapc); \$i++)   {</code>
3	<code>    \$array_id[]= \$listapc['listaPC'][\$i]-&gt;getID();     \$nombreResponsable[]= \$listapc['listaPC'][\$i]-&gt;getResponsable()-</code>
4	<code>    foreach(\$array_id as \$object) {</code>
5	<code>      \$res[] = \$object-&gt;{'\$id'};}</code>
6	<code>    \$result[] = \$servicio-&gt;listarHardwarePCByIdsPC(\$res);</code>
7	<code>  }</code>

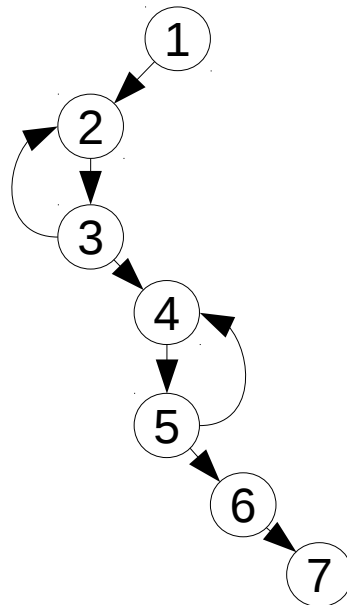


Figura 8: Grafo de Flujo de la funcionalidad createPlantillaLevantamientoHW

Basándose en el grafo de la figura anterior, se tienen 8 aristas y 7 nodos, por lo cual se puede calcular la complejidad ciclomática de la forma siguiente:

$$V(G) = \text{Aristas} - \text{Nodos} + 2$$

$$V(G) = 8 - 7 + 2$$

$$V(G) = 3$$

Del resultado anterior se obtienen 3 caminos, los cuales serían:

- **Camino 1.1:** 1-2-3-4-5-6-7
- **Camino 1.2:** 1-2-3-2-3-4-5-6-7
- **Camino 1.3:** 1-2-3-4-5-4-5-6-7

Para probar los resultados de esta funcionalidad se crean casos de pruebas que garanticen que se ejecute al menos una vez cada sentencia del código, como se muestra en las siguientes tablas.

<b>Caso de Prueba para el Camino 1.1</b>
<b>Descripción de la prueba:</b> Se verifica que se adquiera la información de los dispositivos de hardware
<b>Nombre de la persona que realiza la prueba:</b> Javier Junquera Falcón
<b>Entrada:</b> Se realiza la prueba con una única PC registrada en la entidad con los dispositivos de hardware siguientes: <ul style="list-style-type: none"><li>• <b>Procesador:</b> Intel(R) Pentium(R) III Xeon processor   Velocidad: Mhz</li><li>• <b>Red:</b> Ethernet   TipoMIB: ethernetCsmacd   Descripción: AMD PCNET Family PCI Ethernet Adapter   Velocidad: 100 Mb/s</li><li>• <b>Audio:</b> Intel   Nombre: Intel(r) 82801AA AC"97 Audio Controller   Descripción: Intel(r) 82801AA AC"97 Audio Controller</li><li>• <b>Video:</b> Standard VGA Graphics Adapter   Chipset: VirtualBox VBE BIOS <a href="http://www.virtualbox.org/">http://www.virtualbox.org/</a>  </li></ul>

Memoria: 16MB | Resolución: 1024 x 768

- **Disco:** (Standard disk drives) | Nombre: VBOX HARDDISK | Modelo: //./PHYSICALDRIVE0 | Tipo: Fixed hard disk media | Capacidad: 20473MB
- **Disco:** (Standard CD-ROM drives) | Nombre: VBOX CD-ROM | Modelo: VBOX CD-ROM | Tipo: CD-ROM | Capacidad: 598MB

**Resultado esperado:** Se obtiene una lista con todos los dispositivos de hardware mencionados anteriormente y todos sus datos.

**Resultado obtenido:** Se ha recorrido el Camino 1.1 de forma satisfactoria y se ha obtenido el resultado esperado.

## Caso de Prueba para el Camino 1.2

**Descripción de la prueba:** Se verifica que se adquiera la información de los dispositivos de hardware

**Nombre de la persona que realiza la prueba:** Javier Junquera Falcón

**Entrada:** Se realiza la prueba con más de una PC registrada en la entidad con los dispositivos de hardware siguientes:

### 1. PC# 1

1.1. **Procesador:** Tipo: Intel(R) Pentium(R) III Xeon processor | Velocidad: Mhz

1.2. **Red:** Tipo: Ethernet | TipoMIB: ethernetCsmacd | Descripción: AMD PCNET Family PCI Ethernet Adapter | Velocidad: 100 Mb/s

1.3. **Audio:** Fabricante: Intel | Nombre: Intel(r) 82801AA AC"97 Audio Controller | Descripción: Intel(r) 82801AA AC"97 Audio Controller

1.4. **Video:** Standard VGA Graphics Adapter | Chipset: VirtualBox VBE BIOS <http://www.virtualbox.org/> | Memoria: 16MB | Resolución: 1024 x 768

1.5. **Disco:** Fabricante: (Standard disk drives) | Nombre: VBOX HARDDISK | Modelo: //./

PHYSICALDRIVE0 | Tipo: Fixed hard disk media | Capacidad: 20473MB

1.6. **Disco:** Fabricante: (Standard CD-ROM drives) | Nombre: VBOX CD-ROM | Modelo: VBOX CD-ROM | Tipo: CD-ROM | Capacidad: 598MB

## 2. PC# 2

2.1. **Procesador:** Tipo: Intel Pentium III Xeon processor | Velocidad: MHz

2.2. **Red:** Ethernet | TipoMIB: ethernetCsmacd | Descripción: AMD PCNET Family PCI Ethernet Adapter - Packet Scheduler Miniport | Velocidad: 100 Mb/s

2.3. **Printer:** Modelo: Solid Converter PDF | Controlador: Solid Converter PDF |

2.4. **Video:** Nombre: VirtualBox Graphics Adapter | Chipset: VBOX | Memoria: 16MB | Resolución: 1024 x 665

2.5. **Disco:** Fabricante: (Standard disk drives) | Nombre: VBOX HARDDISK | Modelo: //./ PHYSICALDRIVE0 | Tipo: Fixedxhard disk media | Capacidad: 10236MB

2.6. **Disco:** Fabricante: (Standard CD-ROM drives) | Nombre: VBOX CD-ROM | Modelo: VBOX CD-ROM | Tipo: CD-ROM | Capacidad: 0MB

2.7. **Disco:** Fabricante: (Standard CD-ROM drives) | Nombre: DTSOFT Virtual CdRom Device | Modelo: DTSOFT Virtual CdRom Device | Tipo: CD-ROM | Capacidad: 0MB

**Resultado esperado:** Se obtienen dos listas con todos los dispositivos de hardware mencionados anteriormente y todos sus datos.

**Resultado obtenido:** Se ha recorrido el Camino 1.2 de forma satisfactoria y se ha obtenido el resultado esperado.

Para probar los resultados del camino 1.3 se realiza la misma prueba que para el camino 1.2, o sea, aumentar el número de computadoras en la entidad. Esta prueba arroja los mismos resultados que la prueba anterior.

El siguiente caso de prueba es usando el código fuente del método createPlantillaLevantamientoSW:



Tabla 30: Código de la funcionalidad createPlantillaLevantamientoSW

	<code>public function createPlantillaLevantamientoSW()</code>
	<code>{</code>
1	<code>\$servicio = new nucleo\CoreCapaServicio('ServicioMongo\Hardware');</code> <code>\$listapc = \$servicio-&gt;listarPCs(\$_SESSION['apm']['idProyecto']);</code>
2	<code>for(\$i= 0; \$i&lt;= count(\$listapc); \$i++)</code> <code>{</code>
3	<code>\$array_id[]= \$listapc['listaPC'][\$i]-&gt;getID();</code> <code>}</code>
4	<code>\$servicio = new nucleo\CoreCapaServicio('ServicioMongo\Aplicacion');</code> <code>\$lista_apps[] = \$servicio-&gt;listarAplicacionComputadora(NULL, NULL, \$aplicaciones = array(), \$array_id ,NULL);</code>
5	<code>}</code>

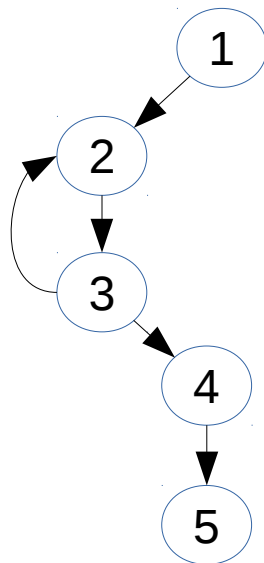


Figura 9: Grafo de Flujo de la funcionalidad createPlantillaLevantamientoSW

Basándose en el grafo de la figura anterior, se tienen 5 aristas y 5 nodos, por lo cual se puede calcular la complejidad ciclomática de la forma siguiente:

$$V(G) = \text{Aristas} - \text{Nodos} + 2$$

$$V(G) = 5 - 5 + 2$$

$$V(G) = 2$$

Del resultado anterior se obtienen 2 caminos, los cuales serían:

- **Camino 2.1:** 1-2-3-4-5
- **Camino 2.2:** 1-2-3-2-3-4-5

Para probar los resultados de esta funcionalidad se crean casos de pruebas que garanticen que se ejecute al menos una vez cada sentencia del código, como se muestra en las siguientes tablas.

<b>Caso de Prueba para el Camino 2.1</b>
<b>Descripción de la prueba:</b> Se verifica que se adquiera la información de las aplicaciones registradas
<b>Nombre de la persona que realiza la prueba:</b> Javier Junquera Falcón
<b>Entrada:</b> Se realiza la prueba con una única PC registrada en la entidad con las aplicaciones siguientes: <ul style="list-style-type: none"><li>• Microsoft Windows Server 2003 Enterprise Edition 5.2.3790 Service Pack 2</li><li>• SMPlayer 0.8.0</li></ul>
<b>Resultado esperado:</b> Se obtiene una lista con todas las aplicaciones mencionados anteriormente y todos sus datos.
<b>Resultado obtenido:</b> Se ha recorrido el camino 2.1 de forma satisfactoria y se ha obtenido el resultado esperado.

<b>Caso de Prueba para el Camino 2.2</b>
<b>Descripción de la prueba:</b> Se verifica que se adquiera la información de las aplicaciones registradas
<b>Nombre de la persona que realiza la prueba:</b> Javier Junquera Falcón
<b>Entrada:</b> Se realiza la prueba con más de una PC registrada en la entidad con las aplicaciones siguientes:

1. PC# 1

1.1. Microsoft Windows Server 2003 Enterprise Edition 5.2.3790 Service Pack 2

1.2. SMPlayer 0.8.0

2. PC# 2

2.1. Complemento Guardar como PDF de Microsoft para programas de Microsoft Office 2007  
12.0.4518.1014

2.2. DAEMON Tools Lite 4.48.1.0347

2.3. Enterprise Architect 7.1 7.0.831

2.4. Equinox Skin Pack Beta-Win8X86 Beta-Win8X86

2.5. FormatFactory 3.2.1.0 3.2.1.0

2.6. Foxit Reader 5.4.5.114

2.7. Google Earth 5.0.11337.1968

**Resultado esperado:** Se obtienen dos listas con todas las aplicaciones mencionados anteriormente y todos sus datos.

**Resultado obtenido:** Se ha recorrido el camino 2.2 de forma satisfactoria y se ha obtenido el resultado esperado.

### 3.2.2 Pruebas de aceptación

Una de las pruebas definidas por la metodología AUP-UCI es la prueba de aceptación. Esta es la prueba final antes del despliegue del sistema. Su objetivo es verificar que el software está listo y que puede ser usado por usuarios finales para ejecutar aquellas funciones y tareas para las cuales el software fue construido [8]. Las pruebas de aceptación se llevaron a cabo para cada una de las Historias de Usuario antes definidas. A continuación se muestran dos de los casos de pruebas realizados al componente implementado.

<b>Caso de Prueba de Aceptación 1</b>	
<b>Código Caso de Prueba:</b> PA-01-01	<b>Nombre Historia de Usuario:</b> Mostrar artefactos a generar
<b>Nombre de la persona que realiza la prueba:</b> Yasiel Pérez Villazón	
<b>Descripción de la Prueba:</b> Probar la interfaz visual del componente	
<b>Condiciones de Ejecución:</b>	
<ul style="list-style-type: none"> <li>• Estar autenticado en la PCMCA.</li> <li>• Que exista al menos un proyecto en la PCMCA.</li> </ul>	
<b>Entrada / Pasos de ejecución:</b>	
<ol style="list-style-type: none"> <li>1. Entrar a la sección Proyectos del subsistema de Migración en la PCMCA.</li> <li>2. Seleccionar un proyecto.</li> <li>3. Ir a la pestaña Expediente</li> </ol>	
<b>Resultado Esperado:</b>	
Se muestra una lista con todos los artefactos a generar.	
<b>Evaluación de la Prueba:</b> Satisfactoria	

<b>Caso de Prueba de Aceptación 2</b>	
<b>Código Caso de Prueba:</b> PA-02-01	<b>Nombre Historia de Usuario:</b> Generar artefacto Levantamiento_hardware
<b>Nombre de la persona que realiza la prueba:</b> Yasiel Pérez Villazón	
<b>Descripción de la Prueba:</b> Probar que se genera el artefacto Levantamiento_hardware con la información existente en la PCMCA correspondiente a este artefacto.	
<b>Condiciones de Ejecución:</b>	
<ul style="list-style-type: none"> <li>• Estar autenticado en la PCMCA.</li> <li>• Que exista al menos un proyecto en la PCMCA.</li> <li>• Que se muestre la lista de los artefactos a generar.</li> </ul>	
<b>Entrada / Pasos de ejecución:</b>	
<ol style="list-style-type: none"> <li>1. Seleccionar el artefacto Levantamiento_hardware.</li> </ol>	
<b>Resultado Esperado:</b>	
Se genera el artefacto Levantamiento_hardware con la información de los dispositivos de hardware existentes en el proyecto seleccionado.	
<b>Evaluación de la Prueba:</b> Satisfactoria	

Al finalizar estas pruebas el cliente fue capaz de verificar el cumplimiento del objetivo planteado. Los

resultados obtenidos para cada uno de los casos de pruebas efectuados fueron satisfactorios. Para medir que el componente desarrollado da solución al problema de la investigación, se comprobó además la satisfacción de los especialistas de migración con el tiempo de realización de los artefactos del Expediente de Proyecto de Migración.

### 3.2.3 Pruebas de satisfacción

Para medir el grado de satisfacción se empleó la técnica de V. A. Iadov, cuya versión original fue creada por su autor para el estudio de la satisfacción por la profesión en carreras pedagógicas [16].

La técnica está conformada por cinco preguntas: tres cerradas y 2 abiertas [17]. Constituye una vía indirecta para el estudio de la satisfacción, ya que los criterios que se utilizan se fundamentan en las relaciones que se establecen entre tres preguntas cerradas que se intercalan dentro de un cuestionario (preguntas 1, 2 y 3 del Anexo 3) y cuya relación el sujeto desconoce. Estas tres preguntas se relacionan a través de lo que se denomina el "Cuadro Lógico de Iadov".

*Tabla 31: Cuadro Lógico de V.A. Iadov [18]  
(con preguntas reformuladas para este caso)*

	¿Prefieres generar los artefactos de forma manual?								
	No			No sé			Sí		
¿Te gusta el componente?	Si pudieras escoger entre usar o no el componente ¿lo usarías?								
	Sí	No sé	No	Sí	No sé	No	Sí	No sé	No
Me gusta mucho	1	2	6	2	2	6	6	6	6
No me gusta tanto	2	2	3	2	3	3	6	3	6
Me da lo mismo	3	3	3	3	3	3	3	3	3
Me disgusta más de lo que me gusta	6	3	6	3	4	4	3	4	4
No me gusta nada	6	6	6	6	4	4	6	4	5

No sé qué decir	2	3	6	3	3	3	6	3	4

El número resultante de la interrelación de las tres preguntas indica la posición de cada sujeto en la escala de satisfacción. La escala de satisfacción es la siguiente:

1 = Clara satisfacción

2 = Más satisfecho que insatisfecho

3 = No definida

4 = Más insatisfecho que satisfecho

5 = Clara insatisfacción

6 = Contradictoria

Para obtener el índice de satisfacción grupal (ISG) se trabaja con los diferentes niveles de satisfacción que se expresan en la escala numérica que oscila entre +1 y - 1 de la siguiente forma:

Tabla 32: Niveles de Satisfacción

1	Máximo de satisfacción
0.5	Más satisfecho que insatisfecho
0	No definido y contradictorio
-0.5	Más insatisfecho que satisfecho
-1	Máxima insatisfacción

La satisfacción grupal se calcula por la siguiente fórmula:

$$ISG = (A(1) + B(0.5) + C(0) + D(-0.5) + E(-1))/N$$

En esta fórmula A, B, C, D, E, representan el número de sujetos con índice individual 1; 2; 3 ó 6; 4; 5 y donde N representa el número total de sujetos del grupo. El índice grupal arroja valores entre + 1 y - 1. Los valores que se encuentran comprendidos entre - 1 y - 0.5 indican insatisfacción; los comprendidos entre - 0,49 y + 0,49 evidencian contradicción y los que caen entre 0.5 y 1 indican que existe satisfacción [18].

En este caso se realizó la encuesta a 13 especialistas de migración, de los cuales 6 respondieron Máxima satisfacción, 5 más satisfechos que insatisfechos y 2 indefinidos.

$$ISG = ( 6 (1) + 5 (0.5) + 2(0) + 0 (-0.5) + 0 (-1)) / 13$$

$$ISG = 0.65$$

Debido a que el valor se encuentra entre 0.5 y 1, esto indica que existe una satisfacción con el componente desarrollado. Con la realización de este componente, una vez se tiene la información necesaria en la PCMCA, se generan los artefactos del expediente con mayor rapidez. Ya no es necesario que los especialistas encargados de generar el Expediente de Proyecto de Migración lo confeccionen de forma manual empleando tiempo ni esfuerzo extra. Se reducen los errores humanos y ya no existe duplicación de la información.

### **3.3 Conclusiones parciales**

En el desarrollo del capítulo, como lo indica la metodología AUP-UCI se realizaron pruebas internas, más específico, la técnica del camino básico del método de caja blanca. Estas pruebas permitieron demostrar el correcto funcionamiento del componente implementado. Además se realizaron las pruebas de aceptación por parte del cliente con resultados satisfactorios, demostrándose el cumplimiento del objetivo y la satisfacción del cliente con la solución implementada. Mediante el uso de la técnica de V. A. Iadov se determinó la satisfacción de los especialistas de migración con el componente desarrollado.

## Conclusiones

La investigación desarrollada y los resultados obtenidos permiten plantear las siguientes conclusiones:

- El análisis de la correspondencia entre el Expediente de Proyecto de Migración y la información recogida por la Plataforma Cubana de Migración a Código Abierto permitió identificar los 10 artefactos del expediente de proyecto que adquieren su información directamente de la PCMCA.
- La utilización de la arquitectura 2-Capas y de las herramientas y tecnologías escogidas, facilitó la integración en el subsistema de Migración del componente implementado.
- La implementación del componente para automatizar la generación de los artefactos del expediente de proyecto disminuye el tiempo de generación de los artefactos del Expediente de Proyecto de Migración.



## **Recomendaciones**

Debido a que el componente genera solamente algunos de los artefactos del Expediente de Proyecto de Migración, se recomienda, en caso de que se agreguen nuevos subsistemas a la PCMCA o funcionalidades nuevas a los subsistemas ya existentes, realizar un nuevo estudio para determinar si se pueden generar nuevos artefactos.

## Bibliografía

- García Vitier, A. y J. García González, «Integración de los sistemas de la Plataforma de Migración a Software Libre y Código Abierto», Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas., Universidad de las Ciencias Informáticas, La Habana, 2011.
- Hernández Leonard, Alejandra Regla. «Redalyc.EVALUACIÓN DE LA SATISFACCIÓN CON EL SERVICIO DE CAPACITACIÓN DEL INIMET - 223028547004.pdf», 2013. [En línea]. Disponible en: <http://www.redalyc.org/pdf/2230/223028547004.pdf>. [Accedido: 09-jun-2016].
- Larman, C. *UML y Patrones. Introducción al análisis y diseño orientado a objetos.*, 2da Edición. 2015. 590 p.
- Lockhart, J. [2015]. *PHP, la manera correcta.* [En línea]. Disponible en: [http://http://librosweb.es/libro/php\\_correcto/capitulo\\_2.html](http://http://librosweb.es/libro/php_correcto/capitulo_2.html)
- López Rodríguez, Alejandro y Viviana González Maura. «La técnica de iadov», abr-2002. [En línea]. Disponible en: <http://www.efdeportes.com/efd47/iadov.htm>. [Accedido: 09-jun-2016].
- Méndez Pérez, J. y Y. C. Abreu Ramírez, «Automatización de la Gestión del Proceso de Migración a Aplicaciones de Código Abierto», Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas., Universidad de las Ciencias Informáticas, La Habana, 2012.
- Pérez Rodríguez, A. Propuesta de expediente de proyecto para la Metodología Cubana de Migración a Software Libre y Código Abierto, Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas., Universidad de las Ciencias Informáticas, La Habana, 2011.
- Pérez Villazón, Y., *Guía cubana de migración de aplicaciones de código abierto.* 2008. p.14
- Rodríguez Sánchez, T., 2015. *Metodología de desarrollo para la Actividad productiva de la UCI.* 3 junio 2015.
- S. Pressman, R. Ingeniería de software. Un enfoque práctico. Quinta edición, 2001. 601 p.

## Referencias Bibliográficas

- [1] Pérez Villazón, Y., *Guía cubana de migración de aplicaciones de código abierto*. 2008. p.14
- [2] Pérez Rodríguez, A. Propuesta de expediente de proyecto para la Metodología Cubana de Migración a Software Libre y Código Abierto, Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas., Universidad de las Ciencias Informáticas, La Habana, 2011.
- [3] Méndez Pérez, J. y Y. C. Abreu Ramírez, «Automatización de la Gestión del Proceso de Migración a Aplicaciones de Código Abierto», Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas., Universidad de las Ciencias Informáticas, La Habana, 2012.
- [4] Reinventando la gestión de datos | MongoDB. [en línea]. [Consulta: 14 noviembre 2015]. Disponible en: <https://www.mongodb.com/es>.
- [5] Potencier, F. y Zaninotto, F., [23 noviembre 2006]. *Symfony 1.4, la guía definitiva*. [en línea]. [Consulta: 3 diciembre 2015]. Disponible en: [http://librosweb.es/libro/symfony\\_1\\_4/capitulo\\_1/symfony\\_en\\_pocas\\_palabras.html](http://librosweb.es/libro/symfony_1_4/capitulo_1/symfony_en_pocas_palabras.html).
- [6] Cochran, D., [12 de noviembre de 2012]. *Twitter Bootstrap Web Development*. 1ra Edición. S.l.: s.n. ISBN 978-1-84951-882-6.
- [7] The PHP Group, 2001-2015. *PHP: ¿Qué es PHP? - Manual*. [en línea]. [Consulta: 16 noviembre 2015]. Disponible en: <http://php.net/manual/es/intro-what-is.php>.
- [8] Rodríguez Sánchez, T., 2015. *Metodología de desarrollo para la Actividad productiva de la UCI*. 3 junio 2015.
- [9] S. Pressman, R. *Ingeniería de software. Un enfoque práctico*. Quinta edición, 2001. 601 p.
- [10] García Vitier, A. y J. García González, «Integración de los sistemas de la Plataforma de Migración a Software Libre y Código Abierto», Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas., Universidad de las Ciencias Informáticas, La Habana, 2011.
- [11] Hill, R., 2007. *Layered Versus Client-Server*. [en línea]. [Consulta: 3 mayo 2016]. Disponible en: <https://msdn.microsoft.com/en-us/library/bb421529.aspx>.
- [12] Larman, C. *UML y Patrones. Introducción al análisis y diseño orientado a objetos*., 2da Edición. 2015. 590 p.
- [13] Lockhart, J. [2015]. *PHP, la manera correcta*. [En línea]. Disponible en: [http://http://librosweb.es/libro/php\\_correcto/capitulo\\_2.html](http://http://librosweb.es/libro/php_correcto/capitulo_2.html)

- [14] «Estandares de programacion. PSR 1 y PSR 4 · jatubio/5minutos\_laravel Wiki · GitHub». [En línea]. Disponible en: [https://github.com/jatubio/5minutos\\_laravel/wiki/Estandares-de-programacion.-PSR-1-y-PSR-4](https://github.com/jatubio/5minutos_laravel/wiki/Estandares-de-programacion.-PSR-1-y-PSR-4). [Accedido: 02-jun-2016].
- [15] «Estandares de programacion. PSR 2 · jatubio/5minutos\_laravel Wiki · GitHub». [En línea]. Disponible en: [https://github.com/jatubio/5minutos\\_laravel/wiki/Estandares-de-programacion.-PSR-1-y-PSR-4](https://github.com/jatubio/5minutos_laravel/wiki/Estandares-de-programacion.-PSR-1-y-PSR-4). [Accedido: 02-jun-2016].
- [16] Kuzmina, N. V. «Metódicas investigativas de la actividad pedagógica». Editorial Leningrado. 1970.
- [17] López Rodríguez, Alejandro. «La importancia de los conocimientos teóricos dentro del sistema de conocimientos de la asignatura Educación Física». Tesis Doctoral. Instituto Central de Ciencias Pedagógicas (ICCP). Ciudad de la Habana. 1993.
- [18] López Rodríguez, Alejandro y Viviana González Maura. «La técnica de Iadov», abr-2002. [En línea]. Disponible en: <http://www.efdeportes.com/efd47/iadov.htm>. [Accedido: 09-jun-2016].

## **Anexos**

### **Anexo 1**

#### **Entrevista realizada al Ing. Yoandy Pérez Villazón, director de CESOL.**

Objetivo: Identificar las empresas en las cuales se ha llevado a cabo el proceso de migración.

#### **Cuestionario**

1. ¿En qué empresas del país el departamento SIMAYS ha realizado el proceso de migración?
2. ¿Quiénes fueron los especialistas involucrados en dicha tarea?

### **Anexo 2**

#### **Entrevista realizada a los especialistas que llevaron a cabo la migración en los OACE.**

Objetivo: Identificar los distintos problemas que trae consigo la generación de forma manual del Expediente de Proyecto de Migración.

#### **Cuestionario**

1. ¿En cuántos procesos de migración que se han llevado a cabo en el departamento SIMAYS ha participado?
2. ¿Cuál ha sido el resultado de dichos procesos?
3. ¿Qué problemas ha traído consigo la generación del Expediente de Proyecto de Migración de forma manual durante el desarrollo de estos procesos?

### **Anexo 3**

#### **Cuestionario para medir grado de satisfacción**

Objetivo: Medir el grado de satisfacción con el componente desarrollado.

Encuesta sobre el componente desarrollado

1-¿Prefieres generar los artefactos de forma manual?

Sí\_\_\_ No\_\_\_ No sé\_\_\_

2- Si pudieras escoger entre usar o no el componente ¿lo usarías?

Sí\_\_\_ No\_\_\_ No sé\_\_\_

3- ¿Te gusta el componente?

Me gusta mucho

No me gusta tanto

Me da lo mismo

Me disgusta más de lo que me gusta

No me gusta nada

No sé qué decir

4- ¿Qué es lo más que le ha gustado del componente?

5- ¿Qué es lo más que le ha disgustado del componente?