

Universidad de las Ciencias Informáticas

Facultad 1



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

*“Componente de soporte técnico para el subsistema de Migración de la
Plataforma Cubana de Migración a Código Abierto”*

Autores:

Jorge Angel Valdés García.

Daniel Romero La Nuez.

Tutores:

Ing. Susana Sánchez Ortiz.

Ing. Danis Carlos Chaviano Jiménez.

La Habana, Junio 2015.

“Año 57 de la Revolución”

Declaración de autoría

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de la Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmamos la presente a los ____ días del mes de _____ del año _____

Firma del autor

Firma del autor

Firma del tutor

Firma del tutor



“Si no existe la organización, las ideas, después del primer momento de impulso, van perdiendo eficacia.”

Ernesto Guevara de la Serna

Dedicatoria

De Daniel:

Este trabajo se lo dedico especialmente a mi heroína de la vida, mi madre por brindarme todo su amor, apoyo, esfuerzo y ser mi ejemplo para todo.

A mis abuelos por estar siempre presentes en cada momento de mi carrera, en especial a mi abuelo Buenaventura por ser mi guía y por sus consejos.

A mi hermana Angela y a Enrique por ser un padre para mí.

A todo el que de una manera u otra estuvo a mi lado estos cinco años apoyándome y aconsejándome.

A mis tutores por su paciencia y su ayuda durante el desarrollo de esto que hoy después de tanto tiempo siendo un gran sueño, ya es una enorme realidad.

De Jorge Angel:

A mi mamá por ser la madre más maravillosa de este mundo y la persona que más amo.

A mis tíos, primos, a mi familia en general por su apoyo incondicional.

A mis tutores por la inmensa ayuda y entrega, por estar siempre dispuestos a apoyarnos en cualquier horario sin ninguna queja.

Agradecimientos

A nuestros padres, familiares y amigos, por apoyarnos a cada instante y ser de mucha ayuda para lograr el cumplimiento de este sueño que hoy se hace realidad.

A nuestro Comandante en Jefe Fidel Castro Ruz y a la revolución cubana por crear esta casa de todos y darnos la oportunidad de formar parte de ella. A nuestros tutores por guiarnos, ayudarnos, confiar en nosotros y estar siempre presente en cada momento que los necesitamos y a todas las personas que de alguna forma u otra han colaborado en nuestra formación como profesionales.

A todas esas personas que durante el transcurso de estos cinco años, han vivido momentos tristes y alegres a nuestro lado. A los que nos ayudaron en la realización de este trabajo.

Resumen

En la presente investigación se muestra el proceso de desarrollo de un componente de soporte técnico para el subsistema de Gestión de la Migración de la Plataforma Cubana de Migración a Código Abierto. Se analizaron diferentes fuentes bibliográficas relacionadas con el soporte técnico, la gestión de incidencias, los niveles de soporte y el proceso de migración a código abierto en Cuba. Además se estudiaron diferentes planes de soporte técnico, dentro de los que destacan, el propuesto por el Centro Nacional de Referencia de Aplicación de las Tecnologías de la Información y las Comunicaciones y la guía de buenas prácticas del Ayuntamiento de Zaragoza, lo que posibilitó la propuesta de un plan de soporte técnico acorde con las necesidades del país. El componente de soporte técnico desarrollado posee como principales funcionalidades la gestión de los mecanismos tecnológicos de apoyo, de los criterios de clasificación de incidencias y de las necesidades de soporte, también permite la clasificación de las incidencias y la gestión del plan de soporte técnico. Las pruebas realizadas demostraron que el componente de soporte técnico contribuye a disminuir el tiempo en la planificación del soporte técnico durante un proceso de migración a código abierto.

Palabras clave: componente, incidencias, migración, plan de soporte, soporte técnico.

Índice de contenido

Introducción.....	1
Capítulo 1: Soporte técnico en la migración a código abierto.....	6
1.1 Soporte técnico.....	6
1.2 Gestión de los servicios de soporte.....	7
1.2.1 Gestión de incidencias.....	8
1.2.2 Tecnologías para la gestión de servicios e incidencias.....	11
1.3 Niveles de soporte.....	12
1.4 Planes de soporte técnico.....	14
1.4.1 CENATIC.....	15
1.4.2 Plan de soporte del Ayuntamiento de Zaragoza.....	17
1.5 Proceso de migración a código abierto.....	19
1.6 Guía cubana de migración a aplicaciones de código abierto.....	20
1.7 Plataforma Cubana de Migración a Código Abierto.....	24
1.7.1 Estructura de la Plataforma Cubana de Migración a Código Abierto.....	25
1.7.2 Flujo de trabajo de la PCMCA.....	27
1.8 Subsistema de Gestión de la Migración.....	28
1.9 Metodología de desarrollo.....	29
1.10 Lenguajes, tecnologías y herramientas.....	30
1.10.1 Lenguajes de programación.....	30
1.10.2 Tecnologías.....	31
1.10.3 Herramientas para el desarrollo.....	31
Conclusiones parciales.....	33
Capítulo 2: Análisis y diseño del componente de soporte técnico para el subsistema de Gestión de la Migración de la PCMCA.....	35
2.1 Propuesta del sistema.....	35
2.1.1 Plan de soporte propuesto.....	37
2.2 Requisitos del componente.....	39
2.2.1 Requisitos funcionales y no funcionales.....	39
2.3 Funcionalidades del componente.....	41
2.4 Arquitectura del componente.....	46
2.5 Diseño del componente.....	47
2.6 Patrones de diseño empleados.....	49
Conclusiones parciales.....	51
Capítulo 3: Implementación y pruebas.....	53
3.1 Plan de liberación.....	53
3.2 Tareas de Ingeniería.....	53

3.3 Pruebas de software.....	56
3.3.1 Pruebas de integración.....	56
3.3.2 Pruebas de aceptación.....	56
3.4 Resultados de las pruebas.....	59
Conclusiones parciales.....	61
Conclusiones generales.....	62
Recomendaciones.....	63
Glosario de términos.....	64
Referencias bibliográficas.....	65

Índice de figuras

Figura 1: Relación entre los componentes de la PCMCA.....	27
Figura 2: Propuesta del sistema.....	36
Figura 3: Modelo de dominio.....	36
Figura 4: Pasos para la elaboración del plan de soporte.....	39
Figura 5: Arquitectura del componente de soporte técnico.....	47
Figura 6: Diagrama de paquetes del componente de soporte técnico.....	48
Figura 7: Resultado de las pruebas de aceptación.....	60

Índice de tablas

Tabla 1: Historia de usuario 1: Gestionar los mecanismos tecnológicos de apoyo.....	42
Tabla 2: Historia de usuario 2: Gestionar criterios de clasificación de incidencias.....	43
Tabla 3: Historia de usuario 3: Gestionar las necesidades de soporte.....	43
Tabla 4: Historia de usuario 4: Clasificar incidencias.....	44
Tabla 5: Historia de usuario 5: Gestionar el plan de soporte.....	45
Tabla 6: Plan de liberación.....	53
Tabla 7: Tarea de ingeniería 1.....	54
Tabla 8: Tarea de ingeniería 2.....	54
Tabla 9: Tarea de ingeniería 3.....	55
Tabla 10: Tarea de ingeniería 4.....	55
Tabla 11: Tarea de ingeniería 5.....	55
Tabla 12: Prueba de aceptación 1.....	57
Tabla 13: Prueba de aceptación 2.....	57
Tabla 14: Prueba de aceptación 3.....	58
Tabla 15: Prueba de aceptación 4.....	59
Tabla 16: Realización de la planificación del soporte técnico sin utilizar el componente implementado.....	61
Tabla 17: Realización de la planificación del soporte técnico utilizando el componente implementado.....	61

Introducción

La creación de las computadoras ha dado paso a un creciente desarrollo tecnológico en las esferas sociales y económicas a nivel mundial. El proceso de informatización en Cuba fue aprobado en el año 1997 con el fin de satisfacer las necesidades de información y conocimiento de la sociedad mediante la utilización masiva de las Tecnologías de la Información y las Comunicaciones (TIC). Sin embargo, como resultado de la política de bloqueo económico y financiero que el gobierno de Estados Unidos mantiene sobre Cuba, resulta insostenible para el país mantenerse a la par del modelo consumista impuesto por las economías capitalistas en el área de las TIC.

Teniendo en cuenta la dependencia absoluta de productos extranjeros y el alto riesgo que implican en la seguridad nacional, surge la necesidad de apostar por la utilización de *software* libre para lograr la soberanía tecnológica del país [1]. En el año 2004 el Consejo de Ministros responde a dicho problema con el Acuerdo 084, que orienta la migración de todos los Organismos de la Administración Central del Estado (OACE) hacia aplicaciones de código abierto. Para la realización de este proceso se crean cuatro grupos de trabajo: legal, capacitación, divulgación y técnico. En la Universidad de las Ciencias Informáticas (UCI) se encuentra el Centro de Software Libre (CESOL), que constituye el eje principal del Grupo Técnico Nacional y que tiene como misión: desarrollar el sistema operativo cubano y conducir procesos de migración a aplicaciones de código abierto.

El departamento encargado de brindar Servicios Integrales en Migración, Asesoría y Soporte (SIMAYS) del centro CESOL elaboró la Guía cubana de migración a aplicaciones de código abierto para conducir los procesos de migración [2]. Este documento define, entre otros elementos, las etapas para la realización de un proceso de migración a código abierto: Preparación, donde se ejecutarán todas las tareas de diagnósticos de los procesos, personas y tecnologías de la entidad, se realizarán tareas de análisis de la información recuperada y se emitirá el plan de migración institucional; Ejecución, comprende las actividades necesarias para la migración definitiva de usuarios y tecnologías de la institución, incluye la migración de los servicios telemáticos y las computadoras de escritorio y finalmente, Consolidación, etapa

que comprende tareas destinadas a garantizar el soporte técnico a los usuarios e infraestructura [3].

A partir de la necesidad de automatizar la mayoría de las actividades realizadas en un proceso de migración a código abierto, se comenzó a desarrollar la Plataforma Cubana de Migración a Código Abierto (PCMCA). Entre sus beneficios incluye la optimización en los tiempos de ejecución de algunas actividades, la no duplicidad de información y la gestión de conocimiento [4].

La PCMCA cuenta con diferentes subsistemas dentro de los que se encuentran: Seguridad; Directorio de software; Certificación y homologación de *hardware*; Inventario de activos tecnológicos; Planificación, control y seguimiento y Gestión de la Migración [4]. Este último posibilita la gestión de la información generada por los restantes subsistemas que conforman la plataforma de migración, permite aumentar la productividad y crea un entorno seguro de almacenamiento, gestión y disponibilidad de los datos [5].

La plataforma, a pesar de automatizar diferentes procesos de la Guía cubana de migración y contar con diferentes subsistemas, no lleva el control del soporte técnico durante un proceso de migración a código abierto. El soporte técnico dentro del proceso de migración a aplicaciones de código abierto es un proceso transversal que comienza su ejecución antes de la migración del primer sistema informático de la entidad. Junto al entrenamiento son los dos procesos fundamentales de cuyo éxito en la realización y calidad dependerán los niveles de aceptación o rechazo de los usuarios a la migración [3].

Para determinar cómo se lleva a cabo el proceso de soporte se realizó una entrevista a especialistas en procesos de migración a código abierto del centro CESOL (ver Anexo 1). En su mayoría los entrevistados plantearon que dicho proceso se realiza utilizando en ocasiones a informáticos de la entidad que se va a migrar, por lo que se hace necesario darle a los mismos una buena capacitación. Además, en las instituciones se recopilan las incidencias manualmente y se hacen llegar al equipo de migración a través del correo, teléfono o de manera presencial. No se evalúa el punto de partida existente, no se define un protocolo de atención a incidencias (las mismas son resueltas según criterio personal), no se definen los mecanismos tecnológicos de apoyo y no se planifican los hitos de soporte. De forma general los especialistas realizan el soporte técnico basados en la experiencia.

Estas carencias ocasionan que el proceso de soporte no se realice de la misma manera en todas las empresas, por lo que se duplican esfuerzos, no se le brinda la correcta atención a los usuarios a la hora de atender sus incidencias y decidir cuál se debe atender primero, y no se reutiliza adecuadamente el conocimiento.

Las insuficiencias indican que se está en presencia del siguiente **problema de investigación**: ¿Cómo disminuir el tiempo de la planificación del soporte técnico durante un proceso de migración a código abierto?

Por lo que se define como **objeto de estudio**: el soporte técnico, enmarcado en el **campo de acción**: el soporte técnico dentro del proceso de migración a código abierto.

El **objetivo general** se centra en desarrollar un componente de soporte técnico para el subsistema de Gestión de la Migración de la Plataforma Cubana de Migración a Código Abierto de forma que se disminuya el tiempo de la planificación del soporte técnico.

Para guiar la investigación se plantea como **idea a defender**:

Con el desarrollo de un componente de soporte técnico para el subsistema de Gestión de la Migración de la Plataforma Cubana de Migración a Código Abierto se contribuirá a disminuir el tiempo de la planificación del soporte técnico durante un proceso de migración a código abierto.

Entre los **objetivos específicos** se encuentran:

1. Caracterizar el soporte técnico dentro del proceso de migración a código abierto.
2. Diseñar e implementar un componente de soporte técnico para el subsistema de Gestión de la Migración de la Plataforma Cubana de Migración a Código Abierto.
3. Probar las funcionalidades presentes en el componente implementado.

Las **tareas de investigación** son las siguientes:

1. Descripción del soporte técnico dentro del proceso de migración a código abierto.
2. Análisis de los planes de soporte existentes.
3. Descripción de la arquitectura de la PCMCA.
4. Definición de los requisitos funcionales y no funcionales a implementar.
5. Diseño e implementación de las clases y métodos que den solución a los requisitos definidos.
6. Diseño y ejecución de casos de prueba al componente implementado.

Para realizar la presente investigación se definieron los siguientes **métodos científicos**:

Métodos teóricos

- Analítico-Sintético: Se utilizó en la revisión de diferentes fuentes bibliográficas y en la extracción de información sobre los temas relacionados con el soporte técnico, los planes de soporte y el soporte técnico dentro del proceso de migración a código abierto.

Métodos empíricos

- Observación: Se empleó para adquirir experiencia y conocimiento durante el proceso de ejecución de un proceso de migración.

- Entrevista: Se le aplicó a los especialistas del Centro de Software Libre de la UCI para obtener información acerca de cómo se lleva a cabo el soporte técnico durante un proceso de migración.

La presente investigación se encuentra estructurada en tres capítulos:

Capítulo 1: Soporte técnico en la migración a código abierto.

Se realiza el análisis y la descripción de los principales conceptos asociados al dominio del problema. Además de la fundamentación teórica relacionada con el soporte técnico y los procesos de migración a código abierto. Se presenta la metodología de desarrollo de *software*, las herramientas a utilizar para la solución a implementar y el lenguaje de modelado.

Capítulo 2: Análisis y diseño del componente de soporte técnico para el subsistema de Gestión de la Migración de la PCMCA.

Se describen los procesos del negocio, se determinan los requisitos funcionales y no funcionales con los que debe cumplir la solución y se describen las historias de usuarios. También se elaboran los diagramas de clases del diseño y se especifica la arquitectura del componente.

Capítulo 3: Implementación y pruebas.

Se define el plan de liberación, las tareas de ingeniería y se realizan las pruebas necesarias para verificar el cumplimiento de los requisitos establecidos.

Capítulo 1: Soporte técnico en la migración a código abierto

El proceso de migración a código abierto que se realiza en las empresas o instituciones debe contener una correcta ejecución del soporte técnico. Para comprender todo lo relacionado con este proceso, en el presente capítulo se analizarán los principales conceptos y características relacionados con el soporte técnico, así como los planes de soporte técnico para ser aplicados durante un proceso de migración. Se realizará una descripción acerca del soporte técnico dentro del proceso de migración a código abierto y de los subsistemas y componentes que conforman la Plataforma Cubana de Migración a Código Abierto. Además se mostrarán las herramientas, metodología y lenguajes de programación que se utilizarán en la implementación del componente.

1.1 Soporte técnico

El soporte técnico es un rango de servicios que proporcionan asistencia con el *hardware* o *software* de una computadora, o algún otro dispositivo electrónico o mecánico. En general, el servicio de soporte técnico sirve para ayudar a resolver los problemas que puedan tener los usuarios, mientras hacen uso de servicios, programas o dispositivos [6].

El soporte técnico, según Afable “es un servicio relacionado a la asistencia a los usuarios que presenten algún problema o incidencias relativas a la tecnología, específicamente a la computadora. Además está relacionado con proveer soluciones aceptables, efectivas y eficientes” [7].

Las compañías e instituciones tienen generalmente soporte técnico interno para empleados, estudiantes y otros asociados. Ofrecen soporte técnico de manera telefónica o en línea y por lo general tienen sus propios empleados. También existen gran cantidad de foros de soporte técnico en Internet que son totalmente gratuitos, y se basan en la simple voluntad y experiencia de los expertos que quieren ayudar a los principiantes [6].

El soporte técnico puede variar dependiendo del rango de posibilidades. Algunos elementos que no son soportados en los niveles bajos de soporte pueden ser soportados en los niveles altos; por ejemplo, las

Capítulo 1: Soporte técnico en la migración a código abierto

preguntas directas pueden ser llevadas a cabo a través de mensajes SMS (*Short Message Service*) o fax; los problemas de *software* básico pueden ser resueltos por teléfono, mientras que los problemas de *hardware* son por lo general tratados en persona [6].

• Tipos de soporte

El soporte técnico se ofrece a través de distintos tipos de medios, incluyendo el correo electrónico, chat, *software* de aplicación, faxes y técnicos de soporte, aunque el más común es el teléfono. En los últimos dos años hay una tendencia a la prestación de soporte técnico en remoto, donde un técnico se conecta al ordenador mediante una aplicación de conexión remota [6].

• Hitos de soporte

Un hito se define como una acción, un acontecimiento o un sujeto que resulta esencial en un cierto contexto, es decir que marca un antes y un después [8]. Un hito de soporte dentro del proceso de migración a código abierto consiste en una fase o etapa en la que se puede dividir el soporte técnico para garantizar el correcto desarrollo del proceso.

1.2 Gestión de los servicios de soporte

Los servicios de soporte se definen como la actividad o función requerida para la conclusión exitosa de un proceso, programa o proyecto. Por otra parte, en la Biblioteca de Infraestructuras de Tecnologías de la Información (ITIL) se define que los servicios de soporte se preocupan de todos los aspectos que garanticen la continuidad, disponibilidad y calidad del servicio prestado al usuario [9].

En los proyectos de desarrollo de *software*, un servicio de soporte amplio y bien gestionado es crítico para obtener éxito con los clientes. Muchas empresas reconocidas en el sector de la informática han implementado soluciones para la gestión del servicio de soporte que ofrecen.

Los servicios de soporte pueden ser utilizados como una forma de retroalimentación, de manera que se puede conocer cómo los clientes están usando el producto, obtener sugerencias para mejoras y determinar prioridades. La responsabilidad de una empresa de desarrollo de *software* con un cliente no

Capítulo 1: Soporte técnico en la migración a código abierto

concluye con la venta del producto. Por lo tanto, un servicio de soporte bien manejado hace que el cliente tenga una fuerte y positiva visión del producto y la empresa que lo desarrolla.

Sin embargo además de estas tareas, también se encargan de gestionar incidencias, capacitar al personal en el uso del *software*, brindar servicios de asistencia técnica ya sea a través de teléfonos de ayuda, sitios web, e incluso de forma presencial, de manera que se garantice la disponibilidad, la continuidad y la calidad del servicio.

- **Gestión de servicios**

La gestión de servicios consiste en “un conjunto de capacidades organizativas especializadas cuyo fin es generar valor a los clientes en forma de servicios” [10]. Por su parte ITIL aborda la gestión de servicios desde un enfoque orientado a entregar servicios de Tecnología de la Información (TI) al cliente en su negocio que alcanzan los objetivos de costo y realización que se marcan en asociación con clientes y englobados en los Acuerdos de Nivel de Servicio (SLA) y Acuerdos de Nivel Operacional (OLA). La gestión de servicios trata de la entrega y apoyo de los servicios de TI que cumplen los requisitos de negocio de la organización, se basa en implementar procesos [11].

1.2.1 Gestión de incidencias

La gestión de incidencias tiene como objetivo resolver, de la manera más rápida y eficaz posible, cualquier incidencia que cause una interrupción en el servicio. La gestión de incidencias no debe confundirse con la gestión de problemas, pues a diferencia de esta última, no se preocupa de encontrar y analizar las causas subyacentes a una determinada incidencia sino exclusivamente de restaurar el servicio. Sin embargo, existe una fuerte interrelación entre ambas [12].

También es importante diferenciar la gestión de incidencias de la gestión de peticiones, que se ocupa de las diversas solicitudes que los usuarios plantean para mejorar el servicio, no cuando este falla [12].

- **Clasificación y Registro**

Capítulo 1: Soporte técnico en la migración a código abierto

Es frecuente que existan múltiples incidencias concurrentes, por lo que es necesario determinar un nivel de prioridad para la resolución de las mismas [13].

La **priorización** se basa esencialmente en dos parámetros [13]:

- **Impacto:** determina la importancia de la incidencia dependiendo de cómo esta afecta a los procesos de negocio y/o del número de usuarios afectados.
- **Urgencia:** depende del tiempo máximo de demora que acepte el cliente para la resolución de la incidencia y/o el nivel de servicio acordado.

También se deben tener en cuenta factores auxiliares tales como el tiempo de resolución esperado y los recursos necesarios: las incidencias “sencillas” se tramitarán cuanto antes; dependiendo de la prioridad, se asignarán los recursos necesarios para la resolución de la incidencia [13].

La prioridad de la incidencia puede cambiar durante su ciclo de vida. Por ejemplo, se pueden encontrar soluciones temporales que restauren aceptablemente los niveles de servicio y que permitan retrasar el cierre de la incidencia sin graves repercusiones. Es conveniente establecer un protocolo para determinar, en primera instancia, la prioridad de la incidencia [13].

• **Escalado y soporte**

Es frecuente que el Centro de Servicios no se vea capaz de resolver en primera instancia una incidencia y para ello deba recurrir a un especialista o a algún superior que pueda tomar decisiones que se escapan de su responsabilidad. A este proceso se le denomina **escalado** [13].

Básicamente hay dos tipos de escalado [13]:

- **Escalado funcional:** se requiere el apoyo de un especialista de más alto nivel para resolver la incidencia.
- **Escalado jerárquico:** se debe acudir a un responsable de mayor autoridad para tomar decisiones que se escapan de las atribuciones asignadas a ese nivel, como por ejemplo,

Capítulo 1: Soporte técnico en la migración a código abierto

asignar más recursos para la resolución de una incidencia específica.

Una correcta gestión de incidencias ofrece como principales beneficios [14]:

- Mejorar la productividad de los usuarios.
- Cumplimiento de los niveles de servicio acordados en el SLA (Acuerdo de Nivel de Servicio).
- Mayor control de los procesos y monitorización del servicio.
- Optimización de los recursos disponibles.
- Una Base de Datos de la Gestión de Configuraciones (CMDB) más precisa pues se registran las incidencias en relación con los elementos de configuración.
- Y principalmente: mejora la satisfacción general de clientes y usuarios.

Sin embargo, una incorrecta gestión de incidencias trae consigo los siguientes efectos adversos [14]:

- Reducción de los niveles de servicio.
- Se derrochan valiosos recursos: demasiadas personas o personas del nivel inadecuado trabajando concurrentemente en la resolución de la incidencia.
- Se pierde valiosa información sobre las causas y efectos de las incidencias para futuras reestructuraciones y evoluciones.
- Se crean clientes y usuarios insatisfechos por la mala y/o lenta gestión de sus incidencias.

El proceso de implementación de la gestión de incidencias puede presentar las siguientes dificultades [14]:

- No se siguen los procedimientos previstos y se resuelven las incidencias sin registrarlas o se escalan innecesariamente y/o omitiendo los protocolos preestablecidos.
- No existe un margen operativo que permita gestionar los “picos” de incidencias por lo que estas no se registran adecuadamente e impiden la correcta operación de los protocolos de

clasificación y escalado.

- No están bien definidos los niveles de calidad de servicio ni los productos soportados. Lo que puede provocar que se procesen peticiones que no se incluían en los servicios previamente acordados con el cliente.

1.2.2 Tecnologías para la gestión de servicios e incidencias

Algunas de las tecnologías para la gestión de servicios e incidencias son los *Service Desk*, *Help Desk* y *Call Center*, las cuales son de suma importancia para la gestión de los servicios e incidencias en las TIC.

Los *Service Desk* (centro de servicio) están compuestos por un conjunto de especialistas de los cuales se espera que tengan la capacidad técnica para contestar cualquier pregunta o queja. Tiene como objetivos proporcionar un punto único de contacto para establecer la comunicación entre la organización y sus clientes, así como gestionar incidencias y peticiones rutinarias de nuevos servicios. La diferencia que presenta ante un *Call Center* (centro de atención de llamadas) o un *Help Desk* (departamento de ayuda) es que tiene un alcance mayor y está más centrado en el cliente, debido a que se encarga de facilitar la integración de los procesos de negocio en la infraestructura TI [9].

Dentro de sus principales procesos está la gestión de incidencias, esto implica que debe estar soportado por la tecnología *Help Desk* que realiza por excelencia este tipo de gestión. Una función importante del *Service Desk* es brindar información a clientes y usuarios relacionada con los nuevos servicios que se brindan, el lanzamiento de nuevas versiones para la corrección de errores, entre otros temas de interés.

Los *Call Center* pueden definirse como el lugar de una empresa donde se concentran las comunicaciones telefónicas de los clientes, o como una unidad o departamento en una empresa que se dedica al cumplimiento de las funciones de comunicación. Es utilizado como centro de atención telefónica de clientes y acciones de *telemarketing*. En ellos se atienden peticiones de información realizadas por el usuario, quejas, entre otros elementos. Disponen de tecnología para automatizar los procesos y tienen capacidad para soportar un gran volumen de llamadas simultáneamente, para atender y mantener el

Capítulo 1: Soporte técnico en la migración a código abierto

registro y redireccionar las llamadas en caso de que sea necesario. Las relaciones que pueden establecerse como un medio de comunicación externa son: entre departamentos en la empresa, relación con usuarios y clientes y funciones de *marketing* [15].

Un *Help Desk* es una parte del grupo de soporte técnico establecido por una organización para mantener operando sus ordenadores en forma eficiente. Un grupo de técnicos de soporte o analistas de *Help Desk* están capacitados para arreglar todo tipo de ordenadores y aplicaciones de *software* que usa dicha organización. Contrario a lo que se pueda pensar, los técnicos no están sentados físicamente ante un escritorio: el *Help Desk* es realmente otro término empleado para denominar al departamento de ayuda. Un *Help Desk* puede ofrecer un rango más amplio de servicios centralizados y ser parte de un centro de servicio más grande [16].

1.3 Niveles de soporte

El soporte o asistencia técnica está a menudo subdividido en capas, o niveles, para que así pueda atender de una forma más eficaz y eficiente a una base de negocio o clientes. El número de niveles en los que una empresa organiza su grupo de soporte depende fundamentalmente de las necesidades del negocio, de los objetivos o de la voluntad, debido a que conllevará a la habilidad para servir de forma suficiente a sus clientes o usuarios [6].

El motivo que justifica prestar un servicio de asistencia a través de un sistema multinivel en lugar de un grupo general de soporte, es proporcionar el mejor servicio posible de la forma más eficiente. El éxito de la estructura organizativa depende enormemente de la capacidad del equipo técnico de comprender su nivel de responsabilidad y compromiso, sus compromisos de tiempo de respuesta al cliente y del momento y forma en la que resulta apropiado escalar una incidencia y hacia qué nivel [6].

De manera general existen 4 niveles de soporte, donde cada uno tiene distintas responsabilidades, estos son: Nivel 0, Nivel 1, Nivel 2 y Nivel 3.

El Nivel 0, también conocido como autoservicio o soporte en línea constituye el primer nivel de soporte. Es el más novedoso debido a que su desarrollo está condicionado al avance de las tecnologías. Proporciona

Capítulo 1: Soporte técnico en la migración a código abierto

al usuario la posibilidad de solucionar sus incidencias por sí mismo, mediante el uso de la ayuda o manuales de usuario, documentación en línea, así como cualquier tipo de documentación técnica que pueda servir como material para el soporte. Este nivel excepcionalmente incluye la comunicación o interacción entre personas como forma de soporte. Muchas compañías con los recursos disponibles para crearlo y mantenerlo, lo implementan [17].

El Nivel 1 es conocido como *Call Center* o *Help Desk* y considerado por muchos como la primera línea de soporte. Consiste en brindar asistencia técnica vía teléfono de manera que se interacciona con los usuarios y/o clientes. Generalmente los usuarios llaman al teléfono de contacto de la empresa, donde son redireccionados al analista de soporte. Este último crea un *ticket*¹ para la incidencia reportada por el usuario en el *software* para el seguimiento de llamadas, donde también registra información del usuario como: nombre, dirección, número telefónico, el problema reportado y una descripción de él. Luego el analista procede a dar solución a la incidencia reportada con la ayuda del conocimiento que ya posee sobre ese tipo de incidencia o apoyado en la documentación del proyecto o en una base de conocimiento. Cuando el analista encuentra solución a la incidencia, lo comunica al usuario que lo reportó, documenta su resolución y cierra el *ticket* [17].

El Nivel 2 es quien maneja el grueso de los *tickets* de soporte y también se le conoce como Soporte *in situ*². En las grandes empresas este nivel se divide en tres grupos de trabajo que son: Soporte en el terreno, *Call Center* y Soporte de *hardware*. Básicamente se encarga de brindar soporte técnico de cara al cliente por lo que se recomienda que los analistas de este nivel tengan de uno a dos años de experiencia dentro de esta área, de manera que hayan creado habilidades en la comunicación con el usuario. Es muy importante la información que el personal de este nivel pueda extraer del usuario respecto a la incidencia, detalles como, qué estaba haciendo antes de que ocurriera la incidencia, pues muchas veces ayuda al entendimiento y a la resolución de la misma por parte del personal de este nivel [17].

1 Resguardo que contiene datos que acreditan ciertos derechos, en la mayoría de los casos obtenidos mediante un pago.

2 Se refiere al soporte con desplazamiento al entorno de los usuarios.

Capítulo 1: Soporte técnico en la migración a código abierto

El Nivel 3 contiene a los siguientes grupos [17]:

- Administradores de Servidores y Bases de Datos.
- Desarrolladores.
- Técnicos en infraestructura de redes.
- Probadores.

Estos grupos son considerados en su conjunto como la última línea de soporte y están conformados por un personal altamente calificado y con habilidades alcanzadas en el trabajo cotidiano en estas áreas de trabajo. Cuando un problema es escalado hasta este nivel, automáticamente estos grupos lo consideran de alta prioridad dado que normalmente afecta a muchos usuarios y porque el *ticket* ya ha estado abierto durante un tiempo considerable, lo que implica que los usuarios probablemente han tenido un tiempo de inactividad del sistema bastante prolongado [17].

1.4 Planes de soporte técnico

Un plan se define como un modelo sistemático de una actuación pública o privada, que se elabora anticipadamente para dirigirla y encauzarla [8].

Los procesos de migración traen asociados un conjunto de problemas y errores no previstos que deben ser rectificadas lo más rápido posible. La solución de estas incidencias se realiza a través del establecimiento de un subproceso de soporte, que debe ser previamente planificado. Se debe establecer un protocolo de atención de incidencias claro y adaptado. Deben establecerse las herramientas necesarias para implementar el protocolo, junto con sus necesidades materiales y de *hardware* [18].

A partir de la información obtenida acerca de los planes de soporte técnico, se encuentran dos planes como los más apropiados para guiar la investigación. Estos son el plan de soporte que propone CENATIC (Centro Nacional de Referencia de Aplicación de las TIC basadas en fuentes abiertas) y el utilizado durante la migración al Ayuntamiento de Zaragoza. Los mismos serán explicados a continuación.

1.4.1 CENATIC

CENATIC es un proyecto estratégico del Gobierno de España creado para impulsar el conocimiento y uso del *software* de código abierto, en todos los ámbitos de la sociedad. La misión del mismo es posicionarse como centro de excelencia nacional, con proyección internacional tanto en el ámbito europeo como iberoamericano [19]. Confecciona su plan de soporte mediante la realización de tres tareas fundamentales:

1) Definir protocolo de atención de incidencias.

En esta tarea se establecerán los flujos de trabajo, las prioridades y los niveles de servicio en base a los elementos prioritarios del escenario origen³ y a la complejidad del escenario destino⁴. Estos elementos, junto con el flujo de estados de las incidencias y el acuerdo del nivel de servicio, compondrán el Protocolo de atención de incidencias [18].

Para poder definir el protocolo de atención a incidencias se establecen las siguientes actividades [18]:

- **Identificar prioridades:**

Se deberán identificar las prioridades que serán asignadas a las incidencias y los criterios de clasificación para cada una de ellas. Se establecerá una prioridad alta para aquellas incidencias que afecten a los elementos críticos. Una vez establecidas las prioridades, se deberán añadir a los criterios de clasificación aquellos casos particulares que afecten a determinados puestos o servicios que previsiblemente necesitarán una prioridad concreta.

- **Definir niveles de servicio:**

Se entiende como Niveles de Servicio a los distintos niveles de tratamiento de incidencias, pudiéndose acordar una calidad de atención diferente para cada uno (tiempo de primera respuesta, tiempo máximo de resolución), en función de la complejidad, esfuerzo necesario para

³ Se define como escenario origen al momento antes de realizar la migración.

⁴ Se define como escenario destino al momento después de la migración.

Capítulo 1: Soporte técnico en la migración a código abierto

su resolución o tipología de la misma.

- **Definir protocolo de atención de incidencias:**

En este paso se definirá el protocolo de atención de incidencias, que incluirá las salidas de los dos pasos anteriores más el Acuerdo de Nivel de Servicio, que establecerá los tiempos máximos de resolución para cada prioridad y nivel de servicio.

Finalmente, se creará el protocolo de atención de incidencias que estará compuesto de las prioridades de atención de incidencias, los niveles de servicio, el acuerdo de nivel de servicio y la lista de estados de incidencias.

2) Definir infraestructura de soporte.

En esta tarea se identifican las herramientas necesarias para poder implementar el protocolo de atención de incidencias y mantener la comunicación necesaria con la entidad y sus usuarios. Se definen las necesidades materiales y de *hardware* de dichas herramientas.

Dentro de las actividades que se deben realizar en esta tarea se encuentran:

- **Definir interfaces de soporte:**

Se deben definir las vías de comunicación que existirán con los usuarios de la entidad (interfaces de soporte), tanto para dar de alta nuevas incidencias como para establecer comunicaciones con los usuarios durante el diagnóstico y resolución.

- **Definir herramientas internas:**

Se definen las herramientas internas que deberán proveer de las suficientes funcionalidades como para dar cumplimiento al protocolo de atención de incidencias. Así mismo, la elección de las herramientas debe tener en cuenta el número de equipos y servicios de la entidad indicados en el escenario origen.

- **Definir necesidades materiales y de *hardware*:**

Capítulo 1: Soporte técnico en la migración a código abierto

Se estudiarán las necesidades materiales y de *hardware* para la puesta en producción de las herramientas elegidas para identificar qué necesidades son requeridas.

3) Planificar soporte.

En esta tarea se unifican las salidas relacionadas con el protocolo de atención de incidencias y las infraestructuras de soporte. Así mismo, se define el período de soporte en base al escenario origen y el escenario destino. Finalmente se elabora el plan de soporte [18].

Para generar el plan de soporte se acordará un período de soporte que será definido en función del escenario origen y el escenario destino. Se establecerá la duración del servicio de soporte y la etapa de la migración en la que se iniciará. Para ello se analizará la complejidad y probabilidad de fallo del *software* elegido en el escenario destino y se combinará con el número de puestos a migrar. Se establecerá también el impacto del período de soporte sobre la gestión del cambio. Finalmente con estos datos, se establece una planificación de soporte por hitos de migración que deberá quedar recogida en el plan de soporte [18].

1.4.2 Plan de soporte del Ayuntamiento de Zaragoza

En el año 2011 en el Ayuntamiento de Zaragoza de España se publica un documento titulado Migración Escritorio Software Libre [19], el cual ofrece una visión global de la información y procesos que se necesitan conocer para poder planificar y realizar una migración con éxito de los entornos de escritorio a herramientas de código abierto. Dentro de los procesos que abarca contiene un plan de soporte, que entre otros elementos refleja los siguientes aspectos [19]:

Mecanismos de reporte de incidencias

En este punto se definen cuáles serán las vías o mecanismos que dispondrán los usuarios para la comunicación de incidencias hacia el servicio técnico, que pueden ser mediante la telefonía, la mensajería instantánea, el correo electrónico o los formularios web.

Sistemas a implementar para proporcionar el soporte

Capítulo 1: Soporte técnico en la migración a código abierto

Estos sistemas se seleccionan en cuanto a la recepción o resolución de la incidencia. La manera de recepcionar la incidencia se realiza vía telefónica o vía formulario web. Y en cuanto a la resolución de la incidencia se tiene en cuenta la actividad presencial de los técnicos o el acceso remoto al ordenador del usuario final mediante la conexión de red.

Niveles de soporte definidos

En función de la dificultad de la incidencia se pueden definir niveles de soporte. Generalmente existe un Nivel 0 que es el que entra en contacto con el usuario y resuelve sus dudas más básicas o reporta hacia otros niveles superiores las incidencias de mayor dificultad.

Por encima del Nivel 0 pueden existir uno o varios niveles en función del volumen y la complejidad de las incidencias reportadas. Es conveniente mantener apartado del Nivel 0 a los técnicos especializados en la migración ya que un alto porcentaje de las consultas de soporte suelen referirse a cuestiones básicas o de formación.

Metodología de resolución de incidencias

Para poder establecer los procesos y procedimientos de soporte para cada organización se necesita la combinación de los mecanismos, sistemas y niveles de soporte. Algunos de los factores que se deben tener en cuenta son:

- Volumen de reportes previstos.
- Horarios de atención necesarios.
- Tiempos de respuestas máximos.
- Prioridades de atención.
- Nivel de cercanía en el soporte (presencial o remoto).
- Disponibilidad de recursos humanos y económicos.

Capítulo 1: Soporte técnico en la migración a código abierto

Reporte y catalogación de las soluciones para uso futuro

Es necesario catalogar y clasificar los tipos de incidencias para que se pueda dedicar más esfuerzos en esa área. Por ejemplo, es habitual que muchas de las consultas recibidas por los técnicos de soporte no hagan referencia a incidencias técnicas si no a cuestiones de uso y manejo del sistema, lo que puede implicar una carencia en otros apartados como el de formación. En la catalogación se obtiene una base de conocimiento que permite almacenar todas las resoluciones anteriores.

A partir de los planes de soporte estudiados anteriormente los autores del presente trabajo dadas las características y necesidades específicas de Cuba, realizan una propuesta de Plan de soporte el cual es descrito en el Capítulo 2 epígrafe 2.1.1.

1.5 Proceso de migración a código abierto

La migración es un proceso complejo de cambios donde toda la institución se ve involucrada. Una serie sucesiva de estos son ejecutados en aras de llevar a la empresa al estado de soberanía e independencia tecnológica, manteniendo índices de productividad aceptables, reduciendo los costos y elevando el nivel de preparación de sus trabajadores [20]. La migración a código abierto posee dos componentes fundamentales: migración social y migración técnica.

La migración social es cuando en el proceso de migración las personas son sometidas a un cambio en su forma de pensar, rompiendo la cultura de fidelización a los programas privativos provocada por la costumbre y adaptándose a la filosofía asociada al movimiento de *software* libre y código abierto [2].

La migración tecnológica es el componente relacionado con el cambio de la tecnología. Su objetivo no es desterrar el *software* privativo de la entidad, sino sustituir aquellos programas que realmente puedan ser cambiados sin afectar sensiblemente el funcionamiento de la institución. Constituye un camino para mejorar la productividad, reducir los costos, crear conocimiento y/o modelos nuevos de negocios. El objetivo principal de la migración técnica es garantizar la compatibilidad entre los programas existentes y sus alternativas [2].

Capítulo 1: Soporte técnico en la migración a código abierto

Por esto se puede decir que la migración no es solo cambiar un sistema operativo por otro o simplemente instalar nuevas aplicaciones, es mucho más que eso, donde el cambio más grande se centra en la forma de pensar de las personas. La migración es un proceso debidamente ordenado donde las personas deben estar convencidas de las ventajas que trae consigo la implantación de nuevas tecnologías y apoyar este proceso desde su comienzo para lograr un desarrollo exitoso del mismo. Para lograr la exitosa ejecución de la migración se hace necesario pasar a través de un conjunto de flujos y etapas descritos en la metodología cubana de migración a plataformas de código abierto, donde se plantean sus principales procesos [20]. Dicha metodología se encuentra contenida en la Guía cubana de migración a aplicaciones de código abierto.

1.6 Guía cubana de migración a aplicaciones de código abierto

Por lo planteado anteriormente y a través de los objetivos declarados para el Grupo Técnico Nacional para la migración en Cuba se crea la Guía cubana de migración a aplicaciones de código abierto. Dicha guía aborda contenidos relacionados con el código abierto dentro de los que se encuentran las definiciones de GNU, Linux y distribución. Se definen las diversas categorías de *software* existentes, tipos de licencias y de estándares informáticos. Cuenta además con un conjunto de anexos importantes para los especialistas que guiarán la migración, incluyéndose en la misma la lista de herramientas alternativas que recomienda el Grupo Técnico Nacional para la migración, organizadas por categorías de *software* [21].

Esta guía cuenta con cuatro capítulos en los que quedan plasmados los principales procesos y etapas para un organizado y eficiente proceso de migración. Los "Aspectos teóricos del Software Libre y el Código Abierto" plantean las principales pautas del porqué de la utilización del *software* libre [1]. La "Migración a tecnologías de Software Libre y Código Abierto" trata temas como: la clasificación de las empresas de acuerdo al nivel de migración y la clasificación de roles involucrados en el proyecto de migración junto a las funciones de cada rol [2]. La "Consultoría en migración a Código Abierto" incluye como principales aspectos: la planificación del diagnóstico, realizar el inventario de *hardware* y *software*, el análisis del negocio y los procesos de la institución, así como la identificación de los interesados [2]. La "Ejecución de la migración" es la etapa que marcará la eliminación o disminución del uso del *software*

Capítulo 1: Soporte técnico en la migración a código abierto

privativo en la institución, la misma estará marcada por tres hitos importantes: la elaboración del plan de migración, la migración de los servicios telemáticos y del escritorio de los usuarios [2].

Actualmente es publicada una reestructuración de dicha guía plasmada en el documento llamado "Buenas Prácticas para la Migración a Código Abierto", esta continuación se basa en las experiencias adquiridas y resultados satisfactorios durante la migración de disímiles organizaciones, tanto en Cuba como en el extranjero, así como en la evolución de las TIC [21].

La reestructuración de la metodología se plantea tras haber detectado un conjunto de dificultades y posibles mejoras en la forma en que se ejecuta dicho proceso, por esta razón surge el problema de cómo perfeccionar el mecanismo existente para la ejecución de los procesos de migración a aplicaciones de código abierto [3]. Esta comprende la reducción del proceso de migración a tres etapas: Preparación, Ejecución y Consolidación, eliminando de esta forma la etapa Migración Parcial que en la práctica nunca se ha podido implementar exitosamente. La correcta planificación del proceso de migración depende mayoritariamente de un exhaustivo levantamiento de información tecnológica, institucional y de recursos humanos que garantice una visión a corto y mediano plazo del estado de la entidad en cuestión. La migración en los servidores y las estaciones de trabajo puede ejecutarse simultáneamente con la restricción de que antes de migrar formatos de archivos deben haberse migrado todos los servicios telemáticos. El uso de herramientas para automatizar varias actividades aporta un considerable ahorro del tiempo y el costo en el proceso de migración [3].

Desde la primera versión de la Guía cubana de migración se han realizado diferentes procesos de migración en Cuba, entre los cuales se pueden mencionar empresas e instituciones como TELEMAR, CECAM y ECOA 24. Durante la consultoría realizada a esta última empresa los especialistas en migración plantearon que el desarrollo correcto de las actividades de soporte técnico permite que la entidad mantenga su funcionamiento normal y cumpla con sus compromisos laborales. Sus funciones son de vital importancia y la efectividad e inmediatez ante la solución de los problemas marcará el grado de éxito de sus labores y de la migración en la entidad [22].

Capítulo 1: Soporte técnico en la migración a código abierto

Sin embargo, el proceso de soporte técnico no se encuentra completamente definido, solo a grandes rasgos; no se tienen en cuenta diferentes elementos que distinguen cada una de las entidades en el proceso de migración. No se define un tiempo de atención a las incidencias, ni una prioridad que permita su clasificación, ni un correcto orden de atención; lo que afecta la organización en el proceso de migración pues se puede incurrir en costos innecesarios [22].

Como resultado de esto se elaboró una estrategia de soporte técnico para la migración a código abierto en Cuba. Esta estrategia está compuesta de cinco etapas fundamentales, las cuales están formadas por un conjunto de acciones a realizar. A continuación se explican cada uno de estas etapas y sus acciones correspondientes.

Etapas de la estrategia de soporte técnico [22]:

1- Evaluación del punto de partida existente.

Lo primero que se debe realizar es un diagnóstico de la institución que se va a migrar, debido a que no todas las instituciones son iguales en cuanto a recursos humanos, capacitación, recursos tecnológicos y otros elementos que definirían el éxito del proceso de soporte y por ende del proceso de migración. Para esto se proponen las siguientes acciones:

- a) Determinar las condiciones de soporte existentes en la institución.
- b) Determinar el nivel de conocimiento de los recursos humanos de la institución en las tecnologías empleadas.
- c) Determinar las condiciones tecnológicas de la institución.

En el caso en que cuenten con un sistema de soporte definido en la institución se debe además:

- d) Valorar el esquema de funcionamiento del sistema actual existente para el soporte y revisar si realmente cumple con los tiempos de respuesta y calidad necesarios.
- e) Revisar el nivel de preparación de los especialistas de soporte técnico, cantidad de especialistas

Capítulo 1: Soporte técnico en la migración a código abierto

contra posibles incidencias que se puedan generar, y calidad del sistema de soporte, términos de conocimiento por parte de los usuarios, calidad de respuesta y nivel de automatización de los procesos.

2- Definición de un protocolo de atención a las incidencias.

Un eficiente mecanismo de atención a incidencias y su cumplimiento posibilitará disminuir los tiempos de respuesta ante las posibles dificultades que puedan enfrentar los usuarios. Para esta etapa se definen las siguientes acciones:

- a) Determinar las prioridades para la atención a las incidencias según las características de la institución.
- b) Clasificar las incidencias de la institución y establecer niveles de prioridad sucesivos.
- c) Definir los niveles para la atención a las incidencias.
- d) Definir el seguimiento y respuesta a las incidencias.

3- Definición de un mecanismo para la gestión de problemas.

La gestión de problemas permite investigar y analizar los problemas que afectan un proceso de migración a código abierto, para identificar causas y proponer soluciones que permitan evitar su repetición. Es por ello que esta etapa consta de las siguientes acciones.

- a) Identificar y registrar los problemas que se presenten en la institución.
- b) Clasificar por prioridad los problemas encontrados.
- c) Elaborar una base de conocimiento con todos los problemas identificados y clasificados para su futura utilización. Esto contribuye a que sea posible la gestión del conocimiento.

4- Definición de los mecanismos tecnológicos de apoyo.

Las tareas de soporte son costosas en cuanto a tiempo y esfuerzo, es por ello que la automatización de

Capítulo 1: Soporte técnico en la migración a código abierto

las mismas es fundamental, se debe definir en dependencia de las condiciones iniciales y del protocolo de atención a incidencias establecido, las herramientas que servirán de apoyo al proceso de soporte en la entidad. Para esta etapa se propone la siguiente acción.

- a) Determinar los mecanismos tecnológicos de apoyo según las características de la institución.

5- Elaboración y aplicación del plan de soporte técnico.

La planificación del soporte técnico es necesaria y debe quedar plasmada en un documento que agrupe lo fundamental del resultado de aplicar cada una de las etapas anteriores y además que sirva de retroalimentación para futuros procesos de migración a código abierto. Esta última etapa se retroalimenta de las etapas anteriores, brindándole cada una de ellas información necesaria que debe contener el plan de soporte. A continuación se muestran las acciones para esta etapa.

- a) Elaborar el plan de soporte técnico.
- b) Aplicar el plan de soporte técnico.

El plan de soporte técnico elaborado debe ser puesto en práctica según los hitos planificados durante el proceso de migración.

1.7 Plataforma Cubana de Migración a Código Abierto

La Plataforma Cubana de Migración a Código Abierto (PCMCA) es un sistema web que permite automatizar la gestión de la información generada en el proceso de migración a código abierto en las diferentes empresas e instituciones involucradas. Está compuesta por un conjunto de subsistemas que interaccionan entre sí en un esquema de servicios, compartiendo datos, funcionalidades y una interfaz de fácil manejo y usabilidad que permite la gestión de los procesos de migración. Por lo que, se logra con este sistema una mayor eficiencia con un menor costo de implementación, con la diferencia que se realiza desde un entorno automatizado y con un control centralizado de la seguridad, caracterizado por una arquitectura donde todos sus sistemas son autónomos y pueden estar disgregados o no físicamente, pero la información y funcionalidades de cada uno es de utilidad para el funcionamiento de los demás [20].

1.7.1 Estructura de la Plataforma Cubana de Migración a Código Abierto

Cada uno de los subsistemas de la PCMCA están definidos e implementados de manera tal que el funcionamiento de los mismos sea factible y con una eficiente interacción entre ellos. A continuación se muestran estos subsistemas y sus funciones principales dentro del sistema.

- Sistema de Gestión de Inventarios de Tecnologías Informáticas (GITI): constituye el sistema encargado de realizar el inventario de *hardware* y *software* de todas las computadoras de la institución. El desarrollo de GITI está basado en la aplicación OCSInventory NG⁵ [21].
- Sistema de gestión de encuestas LimeSurvey⁶: es el sistema que gestiona las encuestas empleadas en el levantamiento de información del personal de la entidad [23].
- Planificación, Control y Seguimiento (PCS): es una solución destinada a la gestión del proyecto de migración que permite administrar las tareas, recursos, entidades y participantes del proyecto de migración a partir de indicadores [24].
- Directorio de Software en Línea (DS): permite el almacenamiento de perfiles de aplicaciones informáticas, relacionándolas por función y cualidades similares. Su principal función es la de realizar búsquedas automatizadas para alternativas de *software* privativo, detectado por GITI durante el proceso de levantamiento y registro de información [25].
- Sistema de Certificación y Homologación de *Hardware* (SCHH): permite el almacenamiento de perfiles de *hardware* y define el nivel de compatibilidad de un dispositivo con distribuciones GNU/Linux [26].
- Subsistema de Seguridad (SCS): es el encargado de proveer las identidades y el control de acceso de manera centralizada a todos los componentes de la PCMCA [21].

⁵ OCSInventory NG: *Open Computer and Software Inventory Next Generation*.

⁶ Aplicación de código abierto para el sistema de encuestas en línea.

Capítulo 1: Soporte técnico en la migración a código abierto

- Sistema para el Manejo de Interfaces (SMI): herramienta para la gestión y manejo de interfaz de usuario de cada sistema dentro de la PCMCA [21].
- Módulo de Gestión de Migración (MGM): es el centro de la plataforma, es quien permite la interacción del resto de los sistemas componentes de la PCMCA [5].
- Subsistema de Formación: permite automatizar la gestión y asignación de cursos.
- Sistema de Seguridad de la PCMCA: se encarga de validar los privilegios de los usuarios y el acceso a los sistemas que conforman la plataforma de migración.
- Manejador de Reportes: posibilita la obtención del contenido necesario de los sistemas que generan información a través de reportes dinámicos [27].

Arquitectura de la PCMCA

La Figura 1 [27], muestra la estructura de los componentes de la PCMCA. Esta arquitectura ayuda a facilitar el proceso de desarrollo por la simplificación del código de las interfaces y a su vez permite la creación de varias interfaces en una misma lógica de negocio [3].

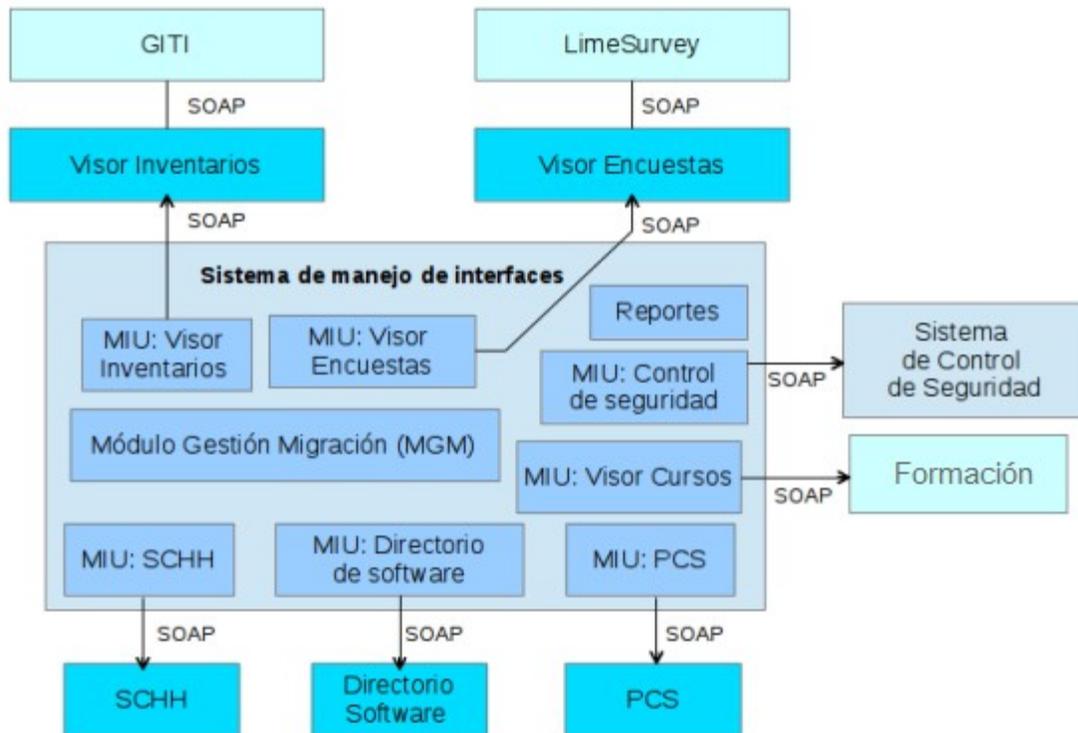


Figura 1: Relación entre los componentes de la PCMCA.

1.7.2 Flujo de trabajo de la PCMCA

Con el objetivo de comprender mejor la función que realiza cada subsistema en la PCMCA, a continuación se describe brevemente el flujo de procesos definidos en la Guía cubana para la migración.

Al iniciarse el proceso de migración en una empresa, el administrador del sistema se autentica e inicia un nuevo proyecto de migración. Esto automáticamente genera una nueva empresa en el sistema PCS, con un proyecto de migración. Según las aplicaciones informáticas instaladas en la institución y detectadas por el subsistema GITI, se crearán encuestas personalizadas para cada usuario a partir de perfiles de conocimiento que se basan en encuestas predeterminadas que tiene el subsistema, las cuales están asociadas a cursos de entrenamiento. Dependiendo de las respuestas de los usuarios arrojadas en las

Capítulo 1: Soporte técnico en la migración a código abierto

encuestas, se podrá conocer el nivel de dominio sobre estas aplicaciones y los cursos de capacitación más adecuados para cada uno de los usuarios. Estas encuestas se registrarán en el sistema de gestión de encuestas *LimeSurvey*. A partir de la información entregada por el subsistema GITI, el subsistema comprueba que el *hardware* detectado esté soportado por GNU/Linux, para ello se chequea la base de datos del SCHH. Se empleará además el DS para buscar las alternativas libres a las aplicaciones privativas detectadas por el subsistema GITI. La plataforma evalúa a petición del usuario, a través del sistema de indicadores definidos en el módulo PCS, en qué por ciento de cumplimiento está el proceso de migración, según las actividades planificadas en el plan de migración, las cuales estarán contenidas como tareas en el módulo PCS [21].

1.8 Subsistema de Gestión de la Migración

El subsistema de Gestión de la Migración es un componente del Sistema Interfaz del Núcleo de la PCMCA, que mediante su relación con los demás subsistemas 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 [5].

El funcionamiento de este subsistema permite la integración de los sistemas que componen la PCMCA mediante la utilización del protocolo SOAP⁷ (*Simple Object Access Protocol*), a su vez permite la automatización de la mayoría de los procesos y actividades identificadas dentro del proceso de migración [5].

La utilización de un gestor de bases de datos no relacional para este subsistema permite optimizar el rendimiento del sistema, teniendo en cuenta la existencia de múltiples llamadas a servicios web que se realizan de forma simultánea a las operaciones del gestor de bases de datos seleccionado. El gestor de base de datos no relacional utilizado en este subsistema brinda una mayor adaptabilidad del sistema a los cambios que pueda sufrir la Metodología cubana de migración [5].

⁷ SOAP: Protocolo que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML [28].

Capítulo 1: Soporte técnico en la migración a código abierto

En el subsistema de Gestión de la Migración se agrupan las funcionalidades en varios componentes: Aplicación, Curso, Encuesta, *Hardware*, Perfil, Proyecto, Red y Regla.

1.9 Metodología de desarrollo

La metodología de desarrollo de *software* constituye un enfoque estructurado que tiene como finalidad hacer más eficaz la producción y lograr alta calidad de una forma costeable. Permite, mediante procedimientos, reglas, técnicas y un soporte documental a desarrolladores, lograr un nuevo *software*. Para el desarrollo de este trabajo de diploma se utilizará la metodología SXP, debido a que es la utilizada por el equipo de proyecto que desarrolla la PCMCA para documentar todos los procesos relacionados con la misma.

SXP es una metodología ágil desarrollada en la UCI, que hace uso de SCRUM⁸ para la gestión del trabajo y toma de XP⁹ las mejores prácticas que guían el desarrollo del *software*, como la refactorización y pruebas continuas. Ofrece una estrategia tecnológica, a partir de la introducción de procedimientos ágiles que permitan actualizar los procesos de *software*, para el mejoramiento de la actividad productiva fomentando el desarrollo de la creatividad, aumentando el nivel de preocupación y responsabilidad de los miembros del equipo, ayudando al líder del proyecto a tener un mejor control del mismo y cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar el éxito del proyecto [31].

SXP consta de 4 fases [31]:

Fase de Planificación-Definición: Se establece la visión, se fijan las expectativas y se realiza el aseguramiento del financiamiento del proyecto. Cuenta con tres tareas principales: ingeniería de sistemas o de información, planificación del proyecto y análisis de los requisitos.

Fase de Desarrollo: Se implementa un sistema listo para entregar en una serie de iteraciones. Cuenta con

8 *Modelo de desarrollo ágil que define un conjunto de prácticas y roles, y que puede tomarse como punto de partida para definir el proceso de desarrollo que se ejecutará durante un proyecto* [29].

9 *eXtreme Programming, metodología de desarrollo de la ingeniería de software que tiene como principal objetivo aumentar la productividad a la hora de desarrollar un proyecto de software* [30].

Capítulo 1: Soporte técnico en la migración a código abierto

tres tareas fundamentales: diseño del *software*, generación de código y prueba del *software*.

Fase de Entrega: Se despliega y se pone en marcha el producto *software*.

Fase de Mantenimiento: Se realiza el soporte para el cliente.

SXP es ideal para proyectos de corta duración con requisitos cambiantes o no bien definidos, donde prevalezca la retroalimentación entre el cliente y el equipo de trabajo. El desarrollo con SXP se realiza en iteraciones cortas (*sprints*), dándole cumplimiento a un grupo de actividades, de las que se generan una serie de artefactos, que documentan el proceso de desarrollo, obteniendo una versión del producto con nuevas funcionalidades [31].

1.10 Lenguajes, tecnologías y herramientas

Para el desarrollo de la presente investigación dentro de los lenguajes de programación, herramientas y tecnologías usadas en la PCMCA se utilizarán *PHP* y *JavaScript* como lenguajes de programación; entre las tecnologías se hará uso de *Symfony* y *jQuery* como marcos de trabajo y *SOAP* como protocolo de comunicación. Las herramientas a utilizar son Netbeans como entorno integrado de desarrollo, como gestor de bases de datos MongoDB, Apache como servidor de aplicaciones web, Visual Paradigm para modelado UML (Lenguaje Unificado de Modelado) y para el control de versiones el Subversion. A continuación se hará un breve resumen de cada uno de los lenguajes, tecnologías y herramientas mencionados.

1.10.1 Lenguajes de programación

PHP (*PHP Hypertext Preprocessor*) es un lenguaje de propósito general muy fácil de aprender con respecto a otros lenguajes utilizados con el mismo fin, como JAVA o ASP. Entre algunas de sus características están: alto rendimiento, gran comunidad de apoyo, multiplataforma, multitud de extensiones y orientado a la web [32].

JavaScript es un lenguaje de programación ligero y orientado a objeto, que permite interactuar con el navegador de manera dinámica y eficaz, proporcionando a las páginas web dinamismo y vida [33].

1.10.2 Tecnologías

Marco de Trabajo

Symfony es un marco de trabajo implementado en PHP 5 para el desarrollo de aplicaciones web. Separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Es compatible con orígenes de bases de datos, entre ellos MySQL, PostgreSQL, Oracle y SQL Server. Facilita la programación de aplicaciones, encapsulando operaciones complejas en instrucciones sencillas. Symfony emplea el tradicional patrón de diseño MVC (modelo-vista-controlador) para separar las distintas partes que forman una aplicación web [34].

jQuery es una biblioteca o *framework* de JavaScript que permite simplificar la manera de interactuar con los documentos HTML (*HyperText Markup Language*), manipular el árbol DOM (*Document Object Model*), manejar eventos, desarrollar animaciones y agregar interacción con la tecnología AJAX (*Asynchronous JavaScript And XML*) a páginas web. Ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código. Entre sus principales características se encuentran la compatibilidad con la mayoría de los navegadores, efectos y animaciones dinámicas, gestión de eventos, manipulación de la hoja de estilos CSS, soporte de extensiones. Tiene una dilatada comunidad de creadores de *plugins* o componentes, lo que hace fácil encontrar soluciones ya creadas en jQuery para implementar interfaces de usuario, galerías, votaciones y efectos diversos [35].

Protocolo de comunicación

SOAP (*Simple Object Access Protocol*) es un protocolo simple de acceso a objetos, basado en XML (*eXtensible Markup Language*) para el intercambio de información entre ordenadores, lo principal del servicio SOAP son los llamados procedimientos remotos transportados a través de HTTP (*Hypertext Transfer Protocol*) [36].

1.10.3 Herramientas para el desarrollo

Entorno integrado de desarrollo

Capítulo 1: Soporte técnico en la migración a código abierto

Netbeans IDE (*Integrated Development Environment*) es un entorno de desarrollo distribuido bajo la doble Licencia CDDL (*Common Development and Distribution License*) y la GPL versión 2 con *Classpath Exception*. Posee soporte para PHP, JavaScript y características visuales para el desarrollo web. Es un producto libre, multiplataforma y gratuito sin restricciones de uso. Sus características son integración nativa con sistemas gestores de bases de datos (MySQL y PostgreSQL), orientación a servicios web y modelado UML, soporte a *frameworks* como Hibernate, Struts, Symfony, Spring entre otros, soporte de CVS (*Control Version System*), soporte para varios lenguajes de programación como PHP, JavaScript, Python y Java [37].

Gestor de Bases de Datos

MongoDB es un sistema de base de datos NoSQL orientado a documentos, en vez de guardar los datos en tablas como se hace en las bases de datos relacionales, guarda la estructura de los datos en documentos tipo JSON (*JavaScript Object Notation*) [38]. Una característica importante que posee es la flexibilidad ya que almacena los datos en los documentos JSON, lo que hace que sea mucho más simple evolucionar su modelo de datos que con un sistema con esquemas forzados como un RDBMS (*Relational database management system*); pues ofrece una gran cantidad de las características de un RDBMS tradicional, tales como los índices secundarios, consultas dinámicas, clasificación, *upserts* (actualizar si el documento existe, insertar en caso contrario), y de agregación. Esto le da la amplitud de la funcionalidad de un RDBMS, con la flexibilidad y capacidad de ampliación que el modelo no relacional permite. MongoDB también ofrece pocas opciones de configuración, y en su lugar trata de hacer de forma automática lo correcto siempre que sea posible, lo que permite concentrarse en el desarrollo de la aplicación, en lugar de emplear tiempo en las configuraciones [39].

Servidor de Aplicaciones Web

Apache surge a partir de un servidor web de la NCSA (*National Center for Super Computing Applications*) como corrección a errores y mejoras importantes al producto inicial. Sus principales características son soporte para el protocolo HTTP, sencillo, con la configuración basada en archivos fáciles de configurar,

Capítulo 1: Soporte técnico en la migración a código abierto

soporte para CGI (*Common Gateway Interface*), soporte de *Host Virtuales*, soporte de autenticación HTTP, soporte de *scripts* PHP [40].

Herramienta de Modelado

Visual Paradigm para UML (VP-UML) es una herramienta de diseño UML y herramienta CASE (*Computer Aided Software Engineering*) UML diseñada para la ayuda al desarrollo de *software*. Ofrece un completo conjunto de herramientas a los equipos de desarrollo de *software* necesario para los requisitos de la captura, *software* de planificación, la planificación de controles, el modelado de clases y modelado de datos [41].

Control de versiones

Subversion es un *software* de sistema de control de versiones diseñado específicamente para reemplazar al popular CVS (*Concurrent Versions System*). Se usa en archivos binarios que, al no poder fusionarse fácilmente, conviene que no sean editados por más de una persona a la vez, se sigue la historia de los archivos y directorios a través de copias y renombrados, y se envían sólo las diferencias en ambas direcciones [42].

Conclusiones parciales

En el desarrollo del capítulo se abordó el análisis y la descripción de los principales conceptos y características del soporte técnico, así como ejemplos de planes de soporte técnico realizados para conducir procesos de migración. La fundamentación teórica sobre los procesos de migración permitió obtener los elementos necesarios relacionados con la realización de dicho proceso para aplicarlo a la implementación de la PCMCA. El estudio de los temas relacionados con las tecnologías para la gestión de servicios e incidencias permitió conocer los conceptos y elementos fundamentales a tener en cuenta para la clasificación y atención de las incidencias. Con el resumen realizado acerca de la Guía cubana de migración y la Plataforma Cubana de Migración a Código Abierto se obtiene la estructura, funcionamiento, herramientas, metodología y lenguajes de programación necesarios para el desarrollo del componente de soporte técnico. El estudio de los planes de soporte permitió determinar los elementos principales a tener

Capítulo 1: Soporte técnico en la migración a código abierto

en cuenta para la propuesta de un plan de soporte técnico propio para Cuba, acorde a las necesidades del país.

Capítulo 2: Análisis y diseño del componente de soporte técnico para el subsistema de Gestión de la Migración de la PCMCA

El soporte técnico constituye uno de los procesos fundamentales dentro de la migración a aplicaciones de código abierto. En el presente capítulo se presenta la propuesta de solución del componente, así como el plan de soporte a implementar para contribuir a mejorar el tiempo de respuesta ante las incidencias detectadas. Se precisan los puntos fundamentales del análisis y diseño del componente como son: el modelo de dominio, las historias de usuario y los patrones de diseño empleados. Además se definen las funcionalidades del componente.

2.1 Propuesta del sistema

Con el estudio realizado en la fundamentación teórica sobre el soporte técnico se demuestra la importancia del mismo dentro de un proceso de migración a código abierto. Por tanto se propone que la PCMCA contenga un componente de soporte técnico dentro del subsistema de Migración, con el objetivo de que el proceso de soporte se realice de la misma manera en todas las empresas, se le brinde la correcta atención a los usuarios a la hora de atender sus incidencias y decidir cuál se debe atender primero, además de reutilizarse adecuadamente el conocimiento. Dicho componente debe permitir gestionar los mecanismos tecnológicos de apoyo, de ellos debe conocerse el nombre, la versión, la licencia, la descripción y las funcionalidades principales. También se deben gestionar los criterios de clasificación de incidencias teniendo en cuenta de estos el nombre, la prioridad, la descripción y el tipo de incidencia. Además, el componente debe permitir gestionar las necesidades de soporte incluyendo los resultados de las encuestas y las características de la institución como son los recursos humanos y el personal capacitado. Contando con estas características se podrán clasificar las incidencias y gestionar los planes de soporte como resultado final.

Capítulo 2: Análisis y diseño del componente de soporte técnico para el subsistema de Gestión de la Migración de la PCMCA

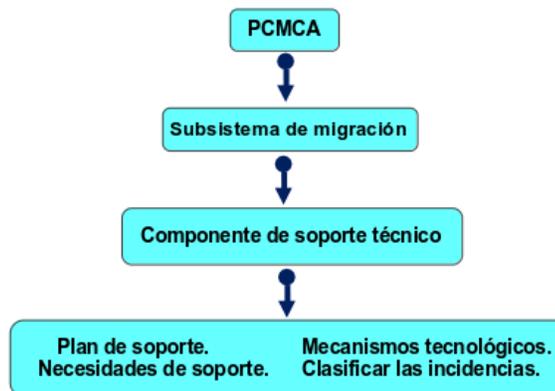


Figura 2: Propuesta del sistema.

En la Figura 2 se muestra de forma general la propuesta de solución, sin embargo se procede a realizar el modelo de dominio para un mejor entendimiento de la solución. Un modelo del dominio se utiliza con frecuencia como fuente de inspiración para el diseño de los objetos *software*. Además muestra las clases conceptuales significativas en un dominio del problema; es el artefacto más importante que se crea durante el análisis orientado a objetos [43].

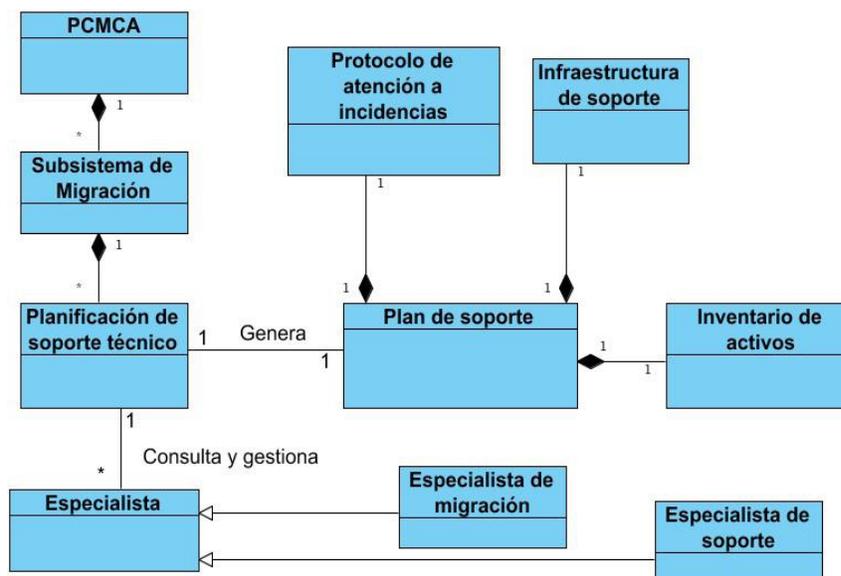


Figura 3: Modelo de dominio.

Capítulo 2: Análisis y diseño del componente de soporte técnico para el subsistema de Gestión de la Migración de la PCMCA

2.1.1 Plan de soporte propuesto

Todo proceso de migración trae asociado un conjunto de problemas y errores no previstos que deben ser solucionados en el menor tiempo posible. La solución de estas incidencias se debe enfrentar a través del establecimiento de un subproceso de soporte, el cual debe ser previamente planificado. Para dar paso a estas soluciones se debe establecer un protocolo de atención de incidencias claro y adaptado tanto al volumen como a la complejidad de la migración. Deben establecerse las herramientas necesarias para la implementación de dicho protocolo, junto con sus necesidades materiales y de *hardware*. Finalmente debe establecerse la duración y el momento de inicio del servicio quedando de esta manera conformado un Plan de soporte para la solución de estas incidencias.

Como resultado de la presente investigación se elabora un Plan de soporte que recopila las condiciones tecnológicas de *hardware* y *software*, recursos humanos y condiciones de soporte con las que cuenta la institución. Teniendo en cuenta que puede ser usado en todas las instituciones sin importar sus características. Mediante el estudio del plan de soporte de CENATIC se abarca como tarea la definición de los protocolos de atención de incidencias, la cual plantea la clasificación de prioridades de atención de incidencias, los niveles de servicio y los diferentes estados por los que pasan dichas incidencias. La infraestructura de soporte es otra de las tareas que en este plan se define, esta contiene los canales de comunicación necesarios para tratar con la entidad y sus usuarios, los requisitos necesarios para montar la infraestructura de soporte y las herramientas para la gestión de incidencias. Como última tarea de este plan está la planificación de soporte donde se plasma el tiempo del servicio de soporte a través de los hitos de migración y el período de soporte definido. Además se tuvieron en cuenta características definidas en el Plan de soporte del Ayuntamiento de Zaragoza como la descripción y clasificación de los reportes de incidencias y las metodologías usadas para la su resolución.

La propuesta del Plan de soporte queda estructurada en cuatro secciones fundamentales. La primera sección se denomina “Necesidades de soporte de la institución”, la misma contendrá las necesidades de

Capítulo 2: Análisis y diseño del componente de soporte técnico para el subsistema de Gestión de la Migración de la PCMCA

soporte de la institución determinadas con anterioridad y estará dividida en tres apartados: recursos humanos, condiciones tecnológicas y sistema de soporte. En la segunda sección se define el protocolo de atención a incidencias, el mismo estará dividido en: prioridades de atención a las incidencias, niveles para la atención a las incidencias y estados de las incidencias. La infraestructura de soporte constituye la tercera sección y estará dividida en: las interfaces de soporte, donde se establecen las vías de comunicación con el usuario; las herramientas de soporte, donde se debe recoger la lista de herramientas de soporte que serán utilizadas, las funcionalidades que cubre cada herramienta y la dificultad estimada para su implantación y uso; y las necesidades materiales y de *hardware*, donde se listan los recursos necesarios para la ejecución del soporte técnico que pueden ser material, servicio y equipamiento informático con la descripción del recurso. La cuarta sección es la planificación del soporte técnico, la misma estará dividida en: hitos de soporte, horario de atención y equipo de soporte.

A continuación se muestra la estructura de dicho Plan de soporte y los pasos para la elaboración del mismo pueden ser consultados en la Figura 4.

Sección 1: Necesidades de soporte de la institución.

1.1. Recursos humanos.

1.2. Condiciones tecnológicas.

1.2.1. *Hardware*.

1.2.2. *Software*.

1.3. Sistema de soporte.

Sección 2: Protocolo de atención a las incidencias.

2.1 Prioridades de atención a las incidencias.

2.2 Niveles para la atención a las incidencias.

Capítulo 2: Análisis y diseño del componente de soporte técnico para el subsistema de Gestión de la Migración de la PCMCA

2.3 Estados de las incidencias.

Sección 3: Infraestructura de soporte.

3.1 Interfaces de soporte.

3.2 Herramientas de soporte.

3.3 Necesidades materiales y de *hardware*.

Sección 4: Planificación del soporte técnico.

4.1 Hitos de soporte.

4.2 Horario de atención.

4.3 Equipo de soporte.

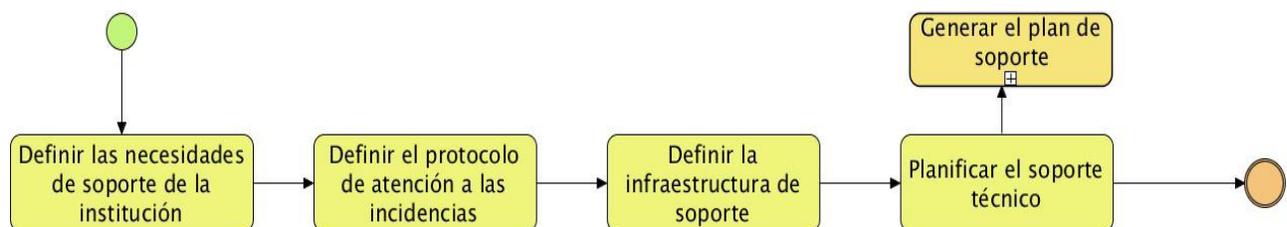


Figura 4: Pasos para la elaboración del plan de soporte.

2.2 Requisitos del componente

Los requisitos incluyen el conjunto de tareas que conducen a comprender cuál será el impacto del *software* sobre el negocio, qué es lo que el cliente quiere y cómo interactúan los usuarios finales con el *software* [43].

2.2.1 Requisitos funcionales y no funcionales

Las funcionalidades a implementar según los requisitos definidos se priorizan según la necesidad y complejidad requerida.

Capítulo 2: Análisis y diseño del componente de soporte técnico para el subsistema de Gestión de la Migración de la PCMCA

Dentro de los requisitos a implementar con una prioridad alta se tienen:

- 1) Adicionar mecanismos tecnológicos de apoyo.
- 2) Eliminar mecanismos tecnológicos de apoyo.
- 3) Editar mecanismos tecnológicos de apoyo.
- 4) Adicionar criterios de clasificación de incidencias.
- 5) Eliminar criterios de clasificación de incidencias.
- 6) Editar criterios de clasificación de incidencias.
- 7) Adicionar necesidades de soporte.
- 8) Eliminar necesidades de soporte.
- 9) Editar necesidades de soporte.
- 10) Clasificar incidencias.
- 11) Adicionar el plan de soporte.
- 12) Eliminar el plan de soporte.
- 13) Editar el plan de soporte.
- 14) Listar el plan de soporte.

Como requisitos de prioridad media están:

- 1) Listar mecanismos tecnológicos de apoyo.
- 2) Listar criterios de clasificación de incidencias.
- 3) Listar necesidades de soporte.

Capítulo 2: Análisis y diseño del componente de soporte técnico para el subsistema de Gestión de la Migración de la PCMCA

Y de prioridad baja se encuentra:

- 1) Exportar el plan de soporte.

Los requisitos no funcionales del componente a implementar son los siguientes:

- 1) Utilizar como lenguajes de programación PHP 5 y JavaScript.
- 2) Utilizar como entorno de desarrollo integrado NetBeans IDE en su versión 8.0.
- 3) Utilizar como herramienta de modelado Visual Paradigm en su versión 10.1.
- 4) Utilizar como servidor de aplicaciones web Apache en su versión 2.2.22.
- 5) Utilizar el marco de trabajo Symfony en su versión 1.4.x para PHP.
- 6) Utilizar el marco de trabajo JQuery para JavaScript.
- 7) Utilizar como mecanismo de intercambio de datos entre aplicaciones el protocolo SOAP.
- 8) Disponer para la instalación de la PCMCA en el SO GNU/Linux de los siguientes paquetes: augeas-lenses, augeas-tools, php5-curl, php5-gd, php5-intl, postgresql, pgadmin3, apache2, php5, php-soap, php5-cgi, php5-dev, php5-cli, php5-pgsql, libapache2-mod-php5, php5-mysql, mysql-server, phpmyadmin, mongoDB.

2.3 Funcionalidades del componente

El componente de soporte técnico incluirá como principales funcionalidades la gestión de los mecanismos tecnológicos de apoyo, la gestión de los criterios de clasificación de incidencias, la gestión de las necesidades de soporte, la clasificación de incidencias y la gestión del plan de soporte. Estas funcionalidades serán descritas a continuación a través de las historias de usuarios.

Las historias de usuario son la técnica utilizada en SXP para especificar los requisitos del *software*, lo que equivale a los casos de uso en el proceso unificado. Las mismas son escritas por los clientes como las

Capítulo 2: Análisis y diseño del componente de soporte técnico para el subsistema de Gestión de la Migración de la PCMCA

tareas que el sistema debe hacer y su construcción depende principalmente de la habilidad que tenga el cliente para definir las. Son escritas en lenguaje natural, no excediendo su tamaño de unas pocas líneas de texto [31].

Tabla 1: Historia de usuario 1: Gestionar los mecanismos tecnológicos de apoyo.

Historia de Usuario	
Número: HU 1	Nombre Historia de Usuario: Gestionar los mecanismos tecnológicos de apoyo.
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Jorge Angel Valdés García.	Iteración Asignada: 1
Prioridad en Negocio: Alta	Puntos Estimados: 5 días
Riesgo en Desarrollo: Alta	Puntos Reales: 5 días
<p>Descripción:</p> <p>Se gestionan los mecanismos tecnológicos de apoyo, que pueden ser para la gestión de incidencias, gestión de problemas, la gestión del conocimiento, la gestión de la comunicación y el soporte remoto. De ellos se incluye el nombre, descripción y tipo de mecanismo. Se almacenan en el sistema, permitiendo adicionar, editar, eliminar y listar cada uno de ellos.</p>	
<p>Observaciones:</p> <p>En la interfaz web del subsistema de Gestión de la Migración:</p> <ol style="list-style-type: none"> 1. Para listar se selecciona en el menú el enlace Migración, submenú Mecanismo Tecnológicos y se muestra un listado de los mecanismos existentes. 2. Para adicionar un mecanismo tecnológico de apoyo son necesarios los siguientes datos: nombre, descripción y tipo de mecanismo. <ul style="list-style-type: none"> • Si existe previamente, se muestra un mensaje de error. 3. Para editar la información de un mecanismo tecnológico, se selecciona el mecanismo, se modifican los datos y posteriormente se actualiza el elemento. 4. Para eliminar un mecanismo tecnológico, se selecciona previamente y se confirma la eliminación. 	
Prototipo de interfaz: Ver Anexo 2.	

Capítulo 2: Análisis y diseño del componente de soporte técnico para el subsistema de Gestión de la Migración de la PCMCA

Tabla 2: Historia de usuario 2: Gestionar criterios de clasificación de incidencias.

Historia de Usuario	
Número: HU 2	Nombre Historia de Usuario: Gestionar criterios de clasificación de incidencias.
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Daniel Romero La Nuez.	Iteración Asignada: 1
Prioridad en Negocio: Alta	Puntos Estimados: 5 días
Riesgo en Desarrollo: Alta	Puntos Reales: 5 días
Descripción:	
Se gestionan los criterios para la clasificación de incidencias. De ellos se incluye el nombre, descripción, capital invertido, frecuencia de uso, paralización de actividades, extensión de uso y dificultad de reemplazo. Se almacenan en el sistema, permitiendo adicionar, editar, eliminar y listar cada una de ellas.	
Observaciones:	
En la interfaz web del subsistema Gestión de la Migración:	
1. Para listar se selecciona en el menú el enlace Migración, submenú Criterios de Clasificación de Incidencias y se muestra un listado de los criterios de clasificación existentes.	
2. Para adicionar un criterio de clasificación de incidencia son necesarios los siguientes datos: nombre, descripción, para el caso de los datos capital invertido, frecuencia de uso, paralización de actividades, extensión de uso y dificultad de reemplazo, se toma un valor por defecto de los mismos, el cual puede ser o no modificado en dependencia de las características de este dato para el criterio de clasificación.	
• Si existe previamente, se muestra un mensaje de error.	
3. Para editar la información de un criterio de clasificación de incidencia, se selecciona, se modifican los datos y posteriormente se actualiza el elemento.	
4. Para eliminar un criterio de clasificación de incidencia, se selecciona previamente y se confirma la eliminación.	
Prototipo de interfaz: Ver Anexo 3.	

Tabla 3: Historia de usuario 3: Gestionar las necesidades de soporte.

Historia de Usuario	
Número: HU 3	Nombre Historia de Usuario: Gestionar las necesidades de soporte.

Capítulo 2: Análisis y diseño del componente de soporte técnico para el subsistema de Gestión de la Migración de la PCMCA

Modificación de Historia de Usuario Número: Ninguna	
Usuario: Daniel Romero La Nuez.	Iteración Asignada: 1
Prioridad en Negocio: Alta	Puntos Estimados: 5 días
Riesgo en Desarrollo: Alta	Puntos Reales: 5 días
<p>Descripción:</p> <p>Se gestionan las necesidades de soporte. Para ello de los proyectos de migración existentes se toman los datos definidos como son recursos humanos, personal capacitado, los resultados de las encuestas, así como necesidades de <i>hardware</i> y <i>software</i> de la institución. Se almacenan en el sistema, permitiendo adicionar, editar, eliminar y listar cada una de ellas.</p>	
<p>Observaciones:</p> <p>En la interfaz web del subsistema Gestión de la Migración:</p> <ol style="list-style-type: none"> 1. Para listar se selecciona en el menú el enlace Migración, submenú Necesidades de Soporte, donde se muestran los proyectos de migración existentes, se selecciona el proyecto deseado y luego se muestran las características y necesidades de la institución. 2. Para adicionar las necesidades de soporte se seleccionan las características listadas de la institución, se selecciona la opción adicionar y se introducen los datos. <ul style="list-style-type: none"> • Si existe previamente, se muestra un mensaje de error. 3. Para editar la información de las necesidades de soporte, se seleccionan los datos, se modifican y posteriormente se actualiza el elemento. 4. Para eliminar las necesidades de soporte, se selecciona previamente y se confirma la eliminación. 	
Prototipo de interfaz: Ver Anexo 4.	

Tabla 4: Historia de usuario 4: Clasificar incidencias.

Historia de Usuario	
Número: HU 4	Nombre Historia de Usuario: Clasificar incidencias.
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Daniel Romero La Nuez.	Iteración Asignada: 2
Prioridad en Negocio: Alta	Puntos Estimados: 5 días
Riesgo en Desarrollo: Alta	Puntos Reales: 5 días

Capítulo 2: Análisis y diseño del componente de soporte técnico para el subsistema de Gestión de la Migración de la PCMCA

Descripción: Se gestionan los elementos para la clasificación de incidencias, para ello es necesario el nombre, descripción, estado y criterios de clasificación, este último haciendo uso de los criterios de clasificación de incidencias definidos. Se almacenan en el sistema, permitiendo adicionar, editar, eliminar y listar cada una de ellas.

Observaciones:

En la interfaz web del subsistema Gestión de la Migración:

1. Para listar se selecciona en el menú el enlace Migración, submenú Clasificación de Incidencias y se muestra un listado de las clasificaciones de incidencias existentes.
2. Para adicionar una clasificación de incidencias son necesarios los siguientes datos: nombre, descripción, estado y criterios de clasificación.
 - Si existe previamente, se muestra un mensaje de error.
3. Para editar la información de una clasificación de incidencias, se seleccionan los datos, se modifican y posteriormente se actualiza el elemento.
4. Para eliminar una clasificación de incidencias, se selecciona previamente y se confirma la eliminación.

Prototipo de interfaz: Ver Anexo 5.

Tabla 5: Historia de usuario 5: Gestionar el plan de soporte.

Historia de Usuario	
Número: HU 5	Nombre Historia de Usuario: Gestionar el plan de soporte.
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Jorge Angel Valdés García.	Iteración Asignada: 2
Prioridad en Negocio: Alta	Puntos Estimados: 6 días
Riesgo en Desarrollo: Alta	Puntos Reales: 6 días
Descripción:	
Se gestionan los planes de soporte a partir de los datos generados por la PCMCA. Se almacenan en el sistema, permitiendo adicionar, editar, eliminar y listar cada uno de ellos. Además se exportan en formato PDF.	
Observaciones:	
En la interfaz web del subsistema Gestión de la Migración:	

Capítulo 2: Análisis y diseño del componente de soporte técnico para el subsistema de Gestión de la Migración de la PCMCA

1. Para listar se selecciona en el menú el enlace Migración, submenú Plan de Soporte y se muestran todos los planes existentes.
2. Para adicionar un plan de soporte son necesarios los datos de la PCMCA recopilados en las historias de usuario anteriores.
 - Si existe previamente, se muestra un mensaje de error.
3. Para editar un plan de soporte se selecciona, se modifican los datos y posteriormente se actualiza el elemento.
4. Para eliminar un plan de soporte, se selecciona previamente y se confirma la eliminación.
5. Para exportar un plan de soporte, se selecciona exportar a PDF y se exporta.

Prototipo de interfaz: Ver Anexo 6.

2.4 Arquitectura del componente

El diseño arquitectónico representa la estructura de datos y los componentes del programa necesarios para construir un sistema computacional. Según Roger Pressman “la arquitectura es la forma en que se integran los diversos componentes del sistema para formar un todo cohesionado. Es la manera en que la solución se amolda y combina con aplicaciones externas. La arquitectura de *software* de un programa es la estructura del sistema que, incluyen los componentes de *software*, las propiedades visibles externamente de esos componentes y las relaciones entre ellos. La arquitectura permite a un ingeniero de *software* analizar la efectividad del diseño para cumplir con los requisitos establecidos, considerar opciones arquitectónicas en una etapa en la que aún resulta relativamente fácil hacer cambios al diseño y reducir los riesgos asociados a la construcción del *software*” [43].

Capítulo 2: Análisis y diseño del componente de soporte técnico para el subsistema de Gestión de la Migración de la PCMCA

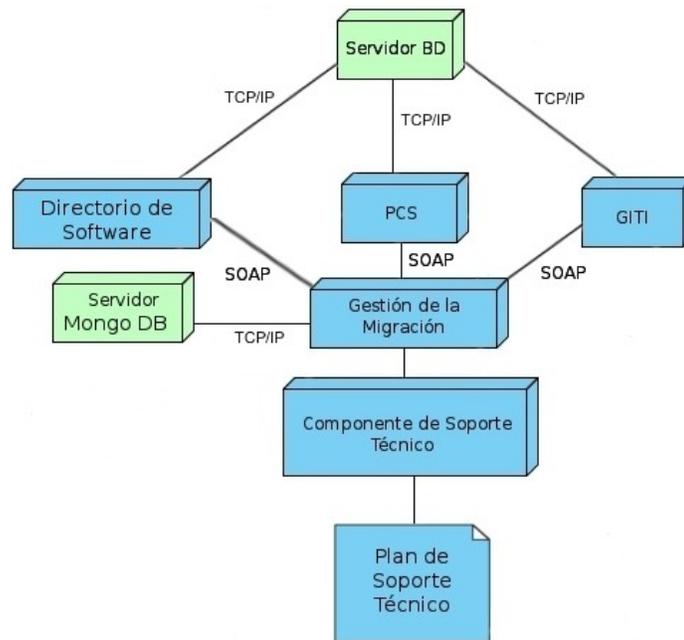


Figura 5: Arquitectura del componente de soporte técnico.

El plan de soporte se confecciona a partir de la información obtenida de los subsistemas GITI, PSC y Gestión de la migración. De GITI se obtienen los activos tecnológicos de la empresa a migrar. PSC contiene los recursos humanos y la planificación de los proyectos. El subsistema de Gestión de la migración es el que incluye el componente de soporte técnico permitiendo la utilización de todas sus funcionalidades. Por otra parte el gestor de bases de datos permite gestionar la información almacenada de manera que se utilice según las necesidades del componente.

2.5 Diseño del componente

En UML un diagrama de paquetes muestra cómo un sistema está dividido en agrupaciones lógicas mostrando las dependencias entre esas agrupaciones. Dado que normalmente un paquete está pensado

Capítulo 2: Análisis y diseño del componente de soporte técnico para el subsistema de Gestión de la Migración de la PCMCA

como un directorio, los diagramas de paquetes suministran una descomposición de la jerarquía lógica de un sistema. Los paquetes están normalmente organizados para maximizar la coherencia interna dentro de cada paquete y minimizar el acoplamiento externo entre los paquetes. La Figura 6 muestra el diagrama de paquetes del componente de soporte técnico [44].

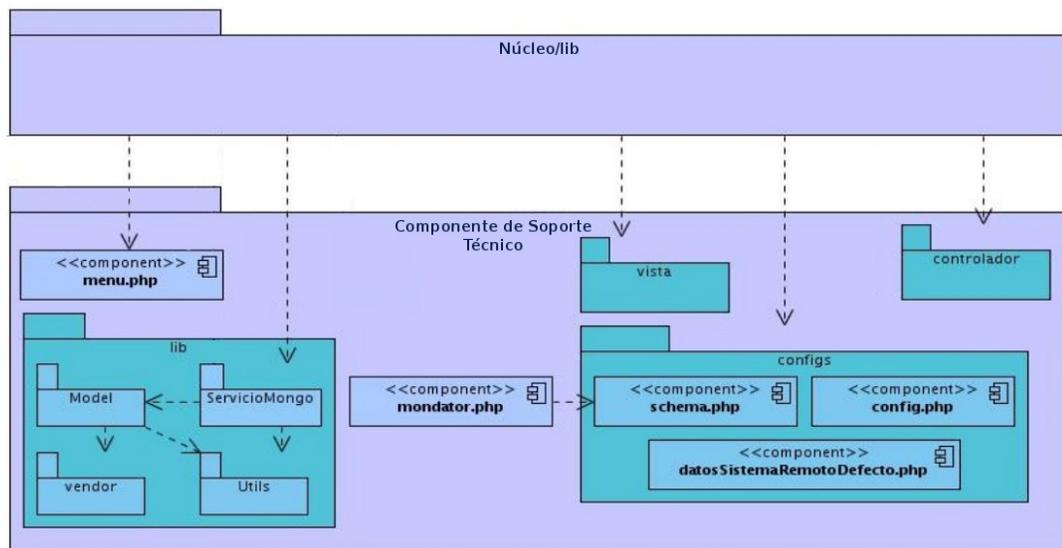


Figura 6: Diagrama de paquetes del componente de soporte técnico.

El paquete Núcleo contiene el sistema de interfaces del núcleo de la PCMCA, el cual está compuesto por un conjunto de clases dentro de las que se encuentran: *CoreAccion*, *CoreCapaServicio*, *CoreServicio*, *CoreVista*, *CoreExporter* y *CorePaginar*. *CoreAccion* provee una interfaz única para la ejecución de las diferentes acciones con los módulos. La clase *CoreServicio* gestiona el acceso a los servicios web. La clase *CorePaginar* provee utilidades para el paginado de las vistas en las que se lista información.

El paquete Componente de soporte técnico contiene los paquetes *vista*, *controlador*, *configs* y *lib*, los cuales son descritos a continuación.

El paquete *vista* contiene las interfaces visuales de las funcionalidades del componente. Su objetivo se basa en mostrar al usuario el funcionamiento de la ejecución de las diferentes acciones descritas a través

Capítulo 2: Análisis y diseño del componente de soporte técnico para el subsistema de Gestión de la Migración de la PCMCA

de una interfaz visual. En el paquete controlador se encuentran todas las clases controladoras del componente las cuales son las encargadas de realizar las acciones de las distintas funcionalidades, entre los que se encuentran: *mechanisms*, *plan*, *criterion*, *incidences*, *needs*. En el paquete *configs* se encuentran los archivos de configuración que permiten el acceso al sistema gestor de bases de datos MongoDB. En el paquete *lib* se encuentran todas las librerías necesarias para utilizar los servicios del gestor de bases de datos MongoDB.

2.6 Patrones de diseño empleados

Un patrón es conocido como “una descripción de un problema y la solución, a la que se da un nombre, y que se puede aplicar a nuevos contextos; idealmente, proporciona consejos sobre el modo de aplicarlo en varias circunstancias, y considera los puntos fuertes y compromisos” [45].

Un patrón arquitectónico especifica un conjunto predefinido de subsistemas con sus responsabilidades y una serie de recomendaciones para organizar los distintos componentes. Resuelve problemas arquitectónicos, de adaptabilidad a requisitos cambiantes, modularidad, acoplamiento, entre otros [45].

Los patrones **GRASP** (*General Responsibility Assignment Software Patterns*) son patrones generales de *software* que describen los principios fundamentales de la asignación de responsabilidades a objetos. El nombre se eligió para indicar la importancia de captar estos principios, con el objetivo de diseñar el *software* de manera eficaz [45]. Dentro de ellos se utilizaron los siguientes [46]:

- Experto: Es un patrón que se usa más que cualquier otro al asignar responsabilidades; es un principio básico que suele ser útil en el diseño orientado a objetos. Con el uso de este patrón se conserva el encapsulamiento, ya que los objetos se valen de su propia información para hacer lo que se les pide. Esto soporta un bajo acoplamiento, lo que favorece al hecho de tener sistemas más robustos y de fácil mantenimiento. Con la utilización de este patrón, se hizo posible definir dónde colocar en cada clase del componente las funcionalidades que necesitan de esa información, dicha clase sería el experto en información. Este patrón se utiliza en las clases

Capítulo 2: Análisis y diseño del componente de soporte técnico para el subsistema de Gestión de la Migración de la PCMCA

controladoras ya que son las expertas en la información.

- Creador: Es un patrón que tiene responsabilidades relacionadas con la creación de objetos. El propósito fundamental del mismo, es encontrar un creador que se debe conectar con el objeto producido en cualquier evento. Este patrón también se utiliza en las clases controladoras ya que crean los formularios que permiten mostrar los campos requeridos en la vista.
- Controlador: Este patrón sugiere que la lógica de negocios debe estar separada de la capa de presentación para aumentar la reutilización de código y a su vez tener un mayor control del flujo de eventos del sistema, se encarga de asignar responsabilidades a clases específicas facilitando la centralización de actividades como la validación, la seguridad, entre otras. Este patrón se utiliza en todo momento debido a que el sistema posee funcionalidades de gestionar. Cada una de estas funcionalidades poseen una clase controladora para guiar la implementación.
- Bajo acoplamiento: El acoplamiento es una medida de la fuerza con que una clase está conectada a otras. Acoplamiento bajo significa que una clase no depende de muchas otras. El uso de este patrón permite que las clases no se afecten por cambios de otros componentes, haciendo posible que sean fáciles de entender y de reutilizar.
- Alta cohesión: La cohesión es una medida de cuán relacionadas y enfocadas están las responsabilidades de una clase. Una alta cohesión caracteriza a las clases con responsabilidades estrechamente relacionadas que no realicen un trabajo enorme. Se hizo necesario la utilización de este patrón en el *software* en cuestión con el fin de controlar la complejidad de cada clase utilizada para mantener un buen comportamiento de las mismas, por esta razón, las que se identificaron con una amplia 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.

Un estilo arquitectónico puede entenderse como un conjunto de principios que definen a alto nivel un

Capítulo 2: Análisis y diseño del componente de soporte técnico para el subsistema de Gestión de la Migración de la PCMCA

aspecto de la aplicación. Este viene definido por un conjunto de componentes, conexiones entre dichos componentes y un conjunto de restricciones sobre cómo se comunican dos componentes cualesquiera una vez conectados [47].

El **estilo de presentación desacoplada** indica cómo debe realizarse el manejo de las acciones del usuario, la manipulación de la interfaz y los datos de la aplicación. Este estilo separa los componentes de la interfaz del flujo de datos y de la manipulación [47].

El **estilo orientado a servicios** permite a una aplicación ofrecer su funcionalidad como un conjunto de servicios para que sean consumidos. Los servicios usan interfaces estándares que pueden ser invocadas, publicadas y descubiertas. Se centra en proporcionar un esquema basado en mensajes con operaciones de nivel de aplicación y no de componente o de objeto [47].

En la implementación inicial de la PCMCA se definen como estilos arquitectónicos SOA (Arquitectura orientada a servicios) y Presentación Desacoplada. La necesidad de una interfaz, desde la cual se pueda acceder a todas las funcionalidades de la plataforma y de que todos sus sistemas componentes intercambiarán información mediante servicios web, evidencia la utilidad de aplicar SOA. Mientras que el uso de la Presentación Desacoplada es evidente al querer simplificar el código de las interfaces con el objetivo de facilitar el proceso de desarrollo y que la plataforma pueda tener varias interfaces para una misma lógica de negocio [47].

Conclusiones parciales

La descripción de la propuesta de solución permitió definir las funcionalidades que debe contener el componente. El análisis y diseño realizado permitió definir los requisitos funcionales y no funcionales, definir y describir las historias de usuario, los patrones de diseño y la arquitectura del componente. Además fue descrito el Plan de soporte propuesto, mostrando las secciones que contiene y los pasos que se deben realizar para su elaboración. De forma general el Plan de soporte recoge la información referente a las necesidades de soporte de la institución, el protocolo de atención a las incidencias, la

Capítulo 2: Análisis y diseño del componente de soporte técnico para el subsistema de Gestión de la Migración de la PCMCA

infraestructura de soporte y los hitos de soporte, lo que posibilita una correcta planificación del soporte técnico durante un proceso de migración.

Capítulo 3: Implementación y pruebas

La etapa de implementación y pruebas es fundamental para el desarrollo de un sistema informático. Durante este capítulo se definen un grupo de artefactos resultantes del proceso de desarrollo de *software* como son: plan de liberación y tareas de ingeniería. También se describen las pruebas de aceptación con el objetivo de encontrar los errores que no habían sido detectados con anterioridad.

3.1 Plan de liberación

Teniendo en cuenta la prioridad asignada para la implementación de las funcionalidades establecidas en las historias de usuarios, se realizaron tres iteraciones para el desarrollo del componente.

Tabla 6: Plan de liberación.

Iteración	Descripción de la iteración	Orden de la HU a implementar	Duración total
1	En esta iteración se implementan todas las historias de usuario que son arquitectónicamente significativas para el cliente, tienen prioridad muy alta y un riesgo de desarrollo medio.	HU 1, HU 2, HU 3	3 semanas
2	En esta iteración se implementan las historias de usuario que tienen prioridad muy alta y un riesgo de desarrollo medio o bajo.	HU 4	3 semanas
3	En esta iteración se implementan las historias de usuario que tienen prioridad muy alta y un riesgo de desarrollo alto.	HU 5	3 semanas

3.2 Tareas de Ingeniería

Las tareas de ingeniería constituyen una definición detallada de las historias de usuarios, facilitando el entendimiento del proceso de implementación. Una historia de usuario puede contener una o más tareas de ingeniería según la necesidad de una buena explicación de las acciones que se realizan en la misma. A

Capítulo 3: Implementación y pruebas

continuación se definen las Tareas de Ingeniería correspondientes a cada una de las Historias de Usuario especificadas en el capítulo anterior [43].

A continuación se muestran las tareas de ingeniería correspondientes a la historia de usuario Gestionar el plan de soporte. Las tareas de ingeniería correspondientes a las restantes historias de usuario se encuentran contenidas en el Anexo 6.

Tabla 7: Tarea de ingeniería 1.

Tarea de ingeniería	
Número Tarea: 1	Número Historia de Usuario: HU 5
Nombre Tarea: Adicionar el plan de soporte.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1 día
Programador Responsable: Jorge Angel Valdés García.	
Descripción: Para adicionar un plan de soporte son necesarios los datos de la PCMCA recopilados en las historias de usuario anteriores. Se recogen las necesidades de soporte, los protocolos de atención y la infraestructura de soporte.	

Tabla 8: Tarea de ingeniería 2.

Tarea de ingeniería	
Número Tarea: 2	Número Historia de Usuario: HU 5
Nombre Tarea: Eliminar el plan de soporte.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1 día
Programador Responsable: Jorge Angel Valdés García.	
Descripción: Para eliminar un plan de soporte, se selecciona previamente y se confirma la	

eliminación.

Tabla 9: Tarea de ingeniería 3.

Tarea de ingeniería	
Número Tarea: 3	Número Historia de Usuario: HU 5
Nombre Tarea: Editar el plan de soporte.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1 día
Programador Responsable: Jorge Angel Valdés García.	
Descripción: Para editar un plan de soporte se selecciona, se modifican los datos y posteriormente se actualiza el elemento.	

Tabla 10: Tarea de ingeniería 4.

Tarea de ingeniería	
Número Tarea: 4	Número Historia de Usuario: HU 5
Nombre Tarea: Listar el plan de soporte.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1 día
Programador Responsable: Jorge Angel Valdés García.	
Descripción: Para listar se selecciona en el menú el enlace Migración, submenú Plan de Soporte y se muestran todos los planes existentes.	

Tabla 11: Tarea de ingeniería 5.

Tarea de ingeniería	
Número Tarea: 5	Número Historia de Usuario: HU 5

Nombre Tarea: Exportar el plan de soporte en formato PDF.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1 día
Programador Responsable: Jorge Angel Valdés García.	
Descripción: Para exportar un plan de soporte, se selecciona exportar a PDF y se exporta.	

3.3 Pruebas de *software*

Las pruebas de *software* son los procesos que permiten verificar la calidad de un producto de *software*. Son utilizadas para identificar posibles fallos de implementación, calidad o usabilidad de un programa. Básicamente es una fase en el desarrollo de *software* con el objetivo de probar las aplicaciones construidas [48].

3.3.1 Pruebas de integración

La prueba de integración es una técnica sistemática para construir la estructura del programa mientras que, al mismo tiempo, se llevan a cabo pruebas para detectar errores asociados con la interacción. El objetivo es tomar los módulos probados mediante las pruebas de aceptación y construir una estructura de programa que esté de acuerdo con lo que dicta el diseño [48]. Mediante la aplicación de la prueba de integración se demostró que el componente de soporte técnico no presenta problemas en la integración con la PCMCA.

3.3.2 Pruebas de aceptación

El objetivo de estas pruebas es verificar los aspectos funcionales del sistema, teniendo en cuenta que las funcionalidades implementadas son la principal fuente de información para comenzar a construir las pruebas de aceptación. Estas constituyen una prueba de caja negra que son creadas a partir de las historias de usuario. A continuación se muestran las pruebas de aceptación correspondientes a la historia

Capítulo 3: Implementación y pruebas

de usuario Gestionar el plan de soporte. Las pruebas de aceptación correspondientes a las restantes historias de usuario se encuentran contenidas en el Anexo 7.

Tabla 12: Prueba de aceptación 1.

Caso de prueba de aceptación	
Código: HU5_P01	Historia de Usuario: Gestionar el plan de soporte.
Nombre: Adicionar el plan de soporte.	
Descripción: Prueba para la funcionalidad adicionar plan de soporte.	
Condiciones de ejecución: El usuario tiene que estar autenticado y poseer los roles de especialista de migración o especialista de soporte.	
Entrada/Pasos de ejecución: Una vez que el usuario se autentica debe ir al submenú Plan de Soporte del subsistema de Gestión de la Migración, se mostrarán los campos requeridos para adicionar un plan. Para adicionar un plan se deben llenar los campos correctamente verificando que no haya sido adicionado con anterioridad.	
Resultado esperado: Al adicionar un plan de soporte correctamente, se actualiza la base de datos con el nuevo plan en la lista. En caso de que algún campo no sea correcto o esté incompleto se muestra un mensaje indicando el error.	
Evaluación de la prueba: Satisfactoria.	

Tabla 13: Prueba de aceptación 2.

Caso de prueba de aceptación	
Código: HU5_P02	Historia de Usuario: Gestionar el plan de soporte.
Nombre: Eliminar el plan de soporte.	
Descripción: Prueba para la funcionalidad eliminar el plan de soporte.	

<p>Condiciones de ejecución: El usuario tiene que estar autenticado y poseer los roles de especialista de migración o especialista de soporte.</p>
<p>Entrada/Pasos de ejecución: Una vez que el usuario se autentica debe ir al submenú Plan de Soporte del subsistema de Gestión de la Migración. Para eliminar un plan, se selecciona y se confirma la eliminación.</p>
<p>Resultado esperado: Al eliminar un plan de soporte correctamente, se actualiza la base de datos con los datos actualizados.</p>
<p>Evaluación de la prueba: Satisfactoria.</p>

Tabla 14: Prueba de aceptación 3.

Caso de prueba de aceptación	
<p>Código: HU5_P03</p>	<p>Historia de Usuario: Gestionar el plan de soporte.</p>
<p>Nombre: Editar el plan de soporte.</p>	
<p>Descripción: Prueba para la funcionalidad editar el plan de soporte.</p>	
<p>Condiciones de ejecución: El usuario tiene que estar autenticado y poseer los roles de especialista de migración o especialista de soporte.</p>	
<p>Entrada/Pasos de ejecución: Una vez que el usuario se autentica debe ir al submenú Plan de Soporte del subsistema de Gestión de la Migración. Para editar la información de un plan, se selecciona, se modifican los datos y posteriormente se actualiza el elemento.</p>	
<p>Resultado esperado: Al editar un plan de soporte correctamente, se actualiza la base de datos con los datos modificados.</p>	
<p>Evaluación de la prueba: Satisfactoria.</p>	

Tabla 15: Prueba de aceptación 4.

Caso de prueba de aceptación	
Código: HU5_P04	Historia de Usuario: Gestionar el plan de soporte.
Nombre: Exportar el plan de soporte.	
Descripción: Prueba para la funcionalidad exportar el plan de soporte.	
Condiciones de ejecución: El usuario tiene que estar autenticado y poseer los roles de especialista de migración o especialista de soporte.	
Entrada/Pasos de ejecución: Una vez que el usuario se autentica debe ir al submenú Plan de Soporte del subsistema de Gestión de la Migración. Para exportar un plan se selecciona exportar a PDF y se exporta.	
Resultado esperado: Un plan de soporte en formato PDF.	
Evaluación de la prueba: Satisfactoria.	

3.4 Resultados de las pruebas.

Las pruebas realizadas han permitido verificar el correcto funcionamiento del componente de soporte para la PCMCA. Demostraron que el componente implementado brinda a los especialistas la posibilidad de crear una infraestructura para la realización del soporte técnico y a su vez lograr una mayor organización y control del desarrollo del soporte. El componente se encarga de la administración de los mecanismos tecnológicos a utilizar en este proceso, también de la creación de criterios de clasificación de incidencias para la posterior clasificación de las mismas, llevando un control de sus principales características. Permite la creación final del plan de soporte técnico que guiará el proceso de soporte durante un proceso de migración y se incluye en el informe de consultoría que es generado en la institución a migrar.

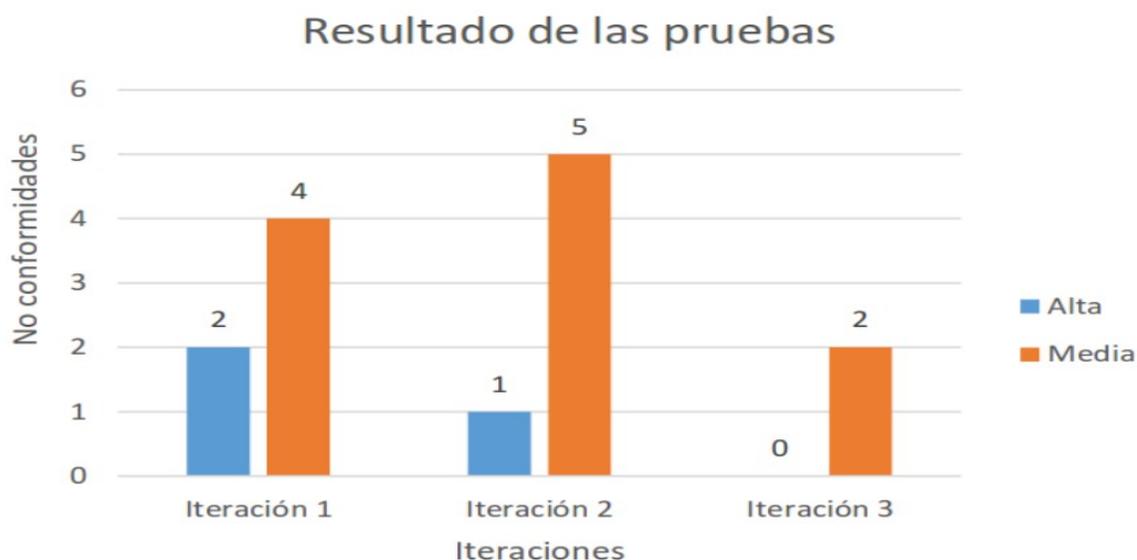


Figura 7: Resultado de las pruebas de aceptación.

Con el desarrollo de cada iteración se le realizaron las pruebas de aceptación al componente de soporte técnico. Como resultado se encontraron 14 no conformidades, tres de prioridad alta y once de prioridad media. Al finalizar cada iteración se le dio solución a las no conformidades detectadas, comenzando por las de prioridad alta, como puede ser observado en la Figura 7.

Para realizar un proceso de migración la PCMCA no contaba con un componente de soporte técnico, por lo que se utilizaba una mayor cantidad de computadoras y un mayor número de personas, y el período de culminación del proceso demoraba más tiempo. A continuación se muestra la Tabla 16 con el tiempo y recursos utilizados en la planificación del soporte técnico sin la utilización del componente de soporte técnico y la Tabla 17 con la utilización del componente implementado. Esta comparación fue realizada a través de la información recopilada de las experiencias y casos de procesos de migración desarrollados por el centro CESOL. La Tabla 16 muestra un promedio del tiempo empleado en la planificación del soporte técnico en varios procesos de migración y la Tabla 17 muestra un caso práctico aplicando el componente en una planificación desarrollada anteriormente.

Capítulo 3: Implementación y pruebas

Tabla 16: Realización de la planificación del soporte técnico sin utilizar el componente implementado.

Tiempo empleado	Recursos utilizados
5 días	7 personas
	2 computadoras

Tabla 17: Realización de la planificación del soporte técnico utilizando el componente implementado.

Tiempo empleado	Recursos utilizados
2 días	1 persona
	1 computadora

La comparación realizada demuestra que con el desarrollo del componente de soporte técnico se disminuye el tiempo de la planificación del soporte técnico; pues sería realizado en solo 2 días y solo sería necesaria una persona (especialista de soporte) y una computadora.

Conclusiones parciales

Con el empleo de la metodología de desarrollo de *software* SXP se generaron las tareas de ingeniería, así como el plan de liberación que sirvió como base para la implementación del componente de soporte técnico. A través de la realización de las pruebas de aceptación se determinó que el componente implementado está en correcto funcionamiento y listo para ser utilizado en la planificación del soporte técnico durante la ejecución de procesos de migración a código abierto. Además quedó demostrado a través de una prueba realizada sobre el tiempo y la utilización de recursos que se empleaban antes y después de la solución, que se contribuye a disminuir el tiempo de la planificación del soporte técnico durante un proceso de migración a código abierto.

Conclusiones generales

- La caracterización del soporte técnico dentro del proceso de migración a código abierto permitió una mayor comprensión de todos los conceptos asociados a soporte técnico, migración a código abierto y planes de soporte.
- El estudio de diferentes planes de soporte permitió la elaboración del Plan de soporte a utilizar para la planificación del soporte técnico durante un proceso de migración a código abierto en Cuba.
- La implementación de un componente de soporte técnico para el Subsistema de Migración posibilita la planificación del soporte técnico desde la Plataforma Cubana de Migración a Código Abierto.
- Las pruebas realizadas a las diferentes funcionalidades del componente de soporte técnico determinaron que el mismo se encuentra correctamente implementado y que contribuye a disminuir el tiempo en la planificación del soporte técnico durante un proceso de migración a código abierto.

Recomendaciones

- Integrar el componente de soporte técnico a la PCMCA y que sea utilizado por los especialistas de soporte técnico para la planificación del soporte técnico durante un proceso de migración a código abierto.
- Agregar al componente de soporte técnico la funcionalidad de gestionar las reglas de clasificación de incidencias para una mayor flexibilidad del mismo.

Glosario de términos

Código abierto: Es el término con el que se conoce al *software* distribuido y desarrollado libremente. El código abierto tiene un punto de vista más orientado a los beneficios prácticos de compartir el código que a las cuestiones morales y/o filosóficas las cuales destacan en el llamado *software* libre.

Incidencia: Circunstancia o suceso secundarios que ocurre en el desarrollo de un asunto o negocio, pero que puede influir en el resultado final.

OLA: Los acuerdos de nivel operacional son acuerdos entre el centro de servicio y otros grupos internos que participan en el servicio de soporte.

PDF: El formato de documento portátil (PDF) es un formato de archivo utilizado para presentar e intercambiar documentos de forma fiable, independiente del *software*, el *hardware* o el sistema operativo.

Prioridad: Hace referencia a la anterioridad de algo respecto de otra cosa, ya que sea en tiempo o en orden.

Priorización: Es una técnica de gestión de la investigación que permite establecer, de acuerdo con unos criterios, un orden de atención.

SLA: Un Acuerdo de Nivel de Servicio es un contrato escrito entre un proveedor de servicio y su cliente con objeto de fijar el nivel acordado para la calidad de dicho servicio.

SXP: Metodología compuesta por las metodologías SCRUM y XP, especialmente indicada para proyectos pequeños, rápido cambio de requisitos donde existe un alto riesgo técnico y se orienta a una entrega rápida de resultados y una alta flexibilidad.

Referencias bibliográficas

1. RODRÍGUEZ, H., GUERRERO, A., PIERRA, A. Y TAMAYO, S.E. El movimiento del Software Libre en Cuba. *Revista Cubana de Ciencias Informáticas*. 2015. Vol. 3, p. 5–12.
2. PÉREZ VILLAZÓN, Yoandy. *Guía cubana de migración a aplicaciones de código abierto*. 2009.
3. PÉREZ VILLAZÓN, YOANDY; GARCÍA VITIER, ABEL; GARCÍA GONZÁLEZ, JAILEN; VIERA HERNÁNDEZ, AMAURY Y OTROS. El proceso de migración a aplicaciones de código abierto en Cuba desde un enfoque metodológico. 2013. Vol. 7, p. 31–41.
4. PÉREZ GUEVARA, DIOSBEL; LEYVA OCHOA, YOENCY; REYES GÓMEZ, LIANNE, PÉREZ VILLAZÓN, YASIEL; MARTÍNEZ FAJARDO, ALEXANDER Y SEGURA RODRÍGUEZ, GRETTEL ARIANNA. PCMCA. . 2014. Vol. 8, p. 78–91.
5. MÉNDEZ PÉREZ, JAVIER Y ABREU RAMÍREZ, JOEL CAMILO. *Automatización de la gestión del proceso de migración a aplicación de código abierto*. Tesis de grado. Habana, Cuba : Universidad de las Ciencias Informáticas, 2012.
6. ANON. El soporte técnico es un rango de servicios. [online]. 2013. Available from: <https://chiquilinom.wordpress.com/2013/05/03/soporte-tecnicoel-soporte-tecnico-es-un-rango-de-servicios-que-proporcionan-asistencia-con/>
7. AFABLE, Jose D. *A Beginner's Guide to Understanding Technical Support* [online]. 2002. Available from: <http://www.amazon.com/Beginners-Guide-Understanding-Technical-Support/dp/0595225748>
8. Diccionario de la lengua española | Real Academia Española. [online]. 2015. [Accessed 29 May 2015]. Available from: <http://www.rae.es/recursos/diccionarios/drae>
9. OSIATIS. Osiatis: Especialistas en Gestión de la Infraestructura de TI. [online]. 2015. Available from: http://itil.osiatis.es/Curso_ITIL/Gestion_Servicios_TI/fundamentos_de_la_gestion_TI/que_es_ITIL/soporte_al_servicio.php.
10. VAN BON, JAN, Y OTROS. *Gestión de servicios TI basado en ITIL V3: Guía de Bolsillo*. 2006. ISBN: 978 90 8753 106 5
11. VILCHES, Ernesto. *Guía de Gestión de Servicios basada en Fundamentos de ITIL v3*. 2010. Luarna Ediciones. ISBN: 978-84-92684-60-1
12. OSIATIS. Gestión de Incidencias Curso ITIL. [online]. 2015. [Accessed 24 May 2015]. Available

Referencias bibliográficas

from: http://itilv3.osiatis.es/operacion_servicios_TI/gestion_incidencias.php

13. OSIATIS. Gestión de Incidencias > Introducción y objetivos > Conceptos básicos. *Gestión de servicios TI* [online]. 2015. [Accessed 24 May 2015]. Available from: http://itilv3.osiatis.es/operacion_servicios_TI/gestion_incidencias/conceptos_basicos.php
14. OSIATIS. Gestión de Incidentes - Introducción y Objetivos. [online]. 2015. [Accessed 24 May 2015]. Available from: http://itil.osiatis.es/Curso_ITIL/Gestion_Servicios_TI/gestion_de_incidentes/introduccion_objetivos_gestion_de_incidentes/introduccion_objetivos_gestion_de_incidentes.php
15. ALONSO, G. *Call Center 100 Success Secrets* [online]. Emereo Pty Limited, 2007. ISBN 9780980459920. Available from: <https://books.google.com.cu/books?id=IltcCTj-cEUC>
16. KNAPP, D. *A Guide to Service Desk Concepts* [online]. Cengage Learning, 2013. ISBN 9781285663340. Available from: <https://books.google.com.cu/books?id=hd0WAAAAQBAJ>
17. FEO GAMIO, Gema. *Modelo técnico de soporte a proyectos informáticos con tecnologías libres*. Tesis de maestría. Villa Clara : Universidad de las Ciencias Informáticas, Centro de desarrollo de software UCI, 2010.
18. ANON. CENATIC. [online]. 2015. [Accessed 24 May 2015]. Available from: <http://www.cenatic.es/>
19. GACIAS MATEO, ALBERTO; GÓMEZ SÁNCHEZ, ALFONSO; ROMERO MORENO, EDUARDO Y PUEYO BENEDICTO, JESÚS. *Migración Escritorio Software Libre*. 2011.
20. PÉREZ VILLAZÓN, YOANDY; GARCÍA VITIER, ABEL Y GARCÍA GONZÁLEZ, JAILEN. *Plataforma Cubana de Migración a Código Abierto*. . 2012. P. 3–4.
21. PÉREZ, Y., GARCÍA, A., GARCÍA, J., HERNÁNDEZ, Y., CUESTA, E., VIERA, A., PÉREZ, A. Y SÁNCHEZ, S. *Buenas prácticas para la migración a código abierto*. Villa Clara : V Taller Nacional Software Libre Presente y Futuro, 2012. ISBN 978-959-250-999-3
22. SÁNCHEZ ORTIZ, SUSANA; PÉREZ BENITEZ, ALFREDO Y GONZÁLEZ HERNÁNDEZ, DELLY LIEN. *Estrategia de soporte técnico para el proceso de migración a software libre en Cuba*. Peña Tecnológica. La Habana : Universidad de las Ciencias Informáticas, 2015.
23. QUINTERO PAJÓN, Mijares Amed. *Módulo de Servicios Web para LimeSurvey*. Tesis de grado. La Habana : Universidad de las Ciencias Informáticas, 2012.
24. GÓMEZ NIVES, MARIYENI Y MARTÍNEZ CALIXTO, LISET. *Sistema de indicadores, planificación y*

Referencias bibliográficas

- seguimiento para la plataforma cubana de migración a Software Libre y Código Abierto*. Tesis de grado. La Habana : Universidad de las Ciencias Informáticas, 2011.
25. ORTÍZ LIMA, YANIRYS Y ALVAREZ ACOSTA, ODAIMY. *Directorio de Software en línea*. Tesis de grado. La Habana : Universidad de las Ciencias Informáticas, 2011.
26. CARBAJO PÉREZ, Ricardo Ramón. *Sistema de Certificación y Homologación de Hardware para Nova GNU/Linux*. Tesis de grado. La Habana : Universidad de las Ciencias Informáticas, 2012.
27. RABELO GARCIA, YOSLEIDY Y AGÜERO VÁZQUEZ, RAMÓN RAFAEL. *Manejador de Reportes para la Plataforma Cubana de Migración a Código Abierto*. Tesis de grado. La Habana : Universidad de las Ciencias Informáticas, 2013.
28. ASHLEY, D. *Simple Object Access Protocol 32 Success Secrets - 32 Most Asked Questions On Simple Object Access Protocol - What You Need To Know* [online]. Emereo Publishing, 2014. ISBN 9781488813757. Available from: <https://books.google.com/cu/books?id=u-UPBwAAQBAJ>
29. RUBIN, K.S. *Essential Scrum: A Practical Guide to the Most Popular Agile Process* [online]. Addison-Wesley, 2012. [Accessed 25 May 2015]. Addison-Wesley signature series. ISBN 9780137043293. Available from: <https://books.google.com/cu/books?id=HkXX65VCZU4C>
30. SILLITTI, A, HAZZAN, O, BACHE, E. and ALBALADEJO, X. *Agile Processes in Software Engineering and Extreme Programming: 12th International Conference, XP 2011, Madrid, Spain, May 10-13, 2011, Proceedings* [online]. Springer, 2011. LNBIP vol 77. ISBN 9783642206764. Available from: <https://books.google.com/cu/books?id=hO2dYE9TFgYC>
31. PEÑALVER ROMERO, GLADYS MARSÍ Y MENESES ABAD, ABEL. *SXP, metodología ágil para proyectos de software libre*. La Habana : Universidad de las Ciencias Informáticas, 2009.
32. HEURTEL, O. *PHP 5.5: Desarrollar un sitio Web dinámico e interactivo* [online]. Ediciones ENI, 2014. Recursos informáticos. ISBN 9782746087989. Available from: <https://books.google.com/cu/books?id=wYozPzHGnsUC>
33. FLANAGAN, D. *JavaScript: The Definitive Guide* [online]. O'Reilly Media, 2011. Definitive Guides. ISBN 9780596805524. Available from: <https://books.google.com/cu/books?id=4RChxt67lvwC>
34. HERNÁNDEZ CARVAJAL, Juan José. *Desarrollo del módulo Administrador de Reportes del Generador Dinámico de Reportes*. Tesis de grado. La Habana : Universidad de las Ciencias Informáticas, 2011.
35. VAN LANCKER, L. *jQuery: El framework JavaScript de la Web 2.0* [online]. Ed. ENI, 2012. Expert IT.

Referencias bibliográficas

ISBN 9782746072589. Available from: <https://books.google.com.cu/books?id=SsTUoWkg588C>

36. ALONSO, G. Web Services. [online]. 2004. [Accessed 11 June 2015]. Available from: http://link.springer.com/chapter/10.1007/978-3-662-10876-5_5

37. HEFFELFINGER, D.R. *Java EE 7 Development with NetBeans 8* [online]. Packt Publishing, 2015. Community experience distilled. ISBN 9781783983537. Available from: <https://books.google.com.cu/books?id=VY92BgAAQBAJ>

38. CORONEL, C. *Bases de Datos, Diseño, Implementacion y Administracion* [online]. Cengage Learning, 2011. ISBN 9786074816181. Available from: https://books.google.com.cu/books?id=KINC0Gc_RREC

39. FRANCIA, S. *MongoDB and PHP* [online]. O'Reilly, 2012. O'Reilly and Associate Series. ISBN 9781449314361. Available from: <https://books.google.com.cu/books?id=Bj--vAEPQDUC>

40. APACHE SOFTWARE FOUNDATION. *Apache HTTP Server 2.2 Official Documentation - Volume I. Server Administration* [online]. Fultus Corporation, 2010. ISBN 9781596821910. Available from: https://books.google.com.cu/books?id=VVCJED_F7QcC

41. OSCAR, S. *Visual Paradigm for Uml* [online]. International Book Market Service Limited, 2013. ISBN 9786139166534. Available from: <https://books.google.com.cu/books?id=PDX-IgEACAAJ>

42. COLLINS-SUSSMAN, B, FITZPATRICK, B and HERNANDEZ, M.T. *Version Control with Subversion* [online]. O'Reilly Media, 2004. O'Reilly Series. ISBN 9780596004484. Available from: <https://books.google.com.cu/books?id=v1rN2MJ81JUC>

43. PRESSMAN, R.S. *Software Engineering: A Practitioner's Approach* [online]. Boston, 2005. McGraw-Hill series in computer science. ISBN 9780073019338. Available from: <https://books.google.com.cu/books?id=bL7QZHtWvaUC>

44. Diagramas de paquetes. [online]. 2015. [Accessed 4 June 2015]. Available from: <http://es.slideshare.net/moyscruz/diagramas-de-paquetes>

45. LARMAN, C. *UML y Patrones. Una introducción al análisis y diseño orientado a objetos y al proceso unificado*. 2003. ISBN 970-17-0261-1

46. HERNANDEZ, M.T. *Symfony Framework: Desarrollo Rápido de Aplicaciones Web* [online]. CreateSpace, 2015. [Accessed 25 May 2015]. ISBN 9781511755078. Available from: https://books.google.com.cu/books?id=RgE_rgEACAAJ

Referencias bibliográficas

47. GARCÍA VITIER, ABEL Y GARCÍA GONZÁLEZ, JAILEN. *Integración de los sistemas de la Plataforma de Migración a Software Libre y Código Abierto*. Tesis de grado. La Habana : Universidad de las Ciencias Informáticas, 2011.
48. CHAVIANO JIMÉNEZ, Danis Carlos. *Herramienta informática para el monitoreo de errores de las aplicaciones web en el centro Fortes*. Tesis de grado. La Habana : Universidad de las Ciencias Informáticas, 2014.