

**Universidad de las Ciencias Informáticas  
Facultad 1**



**Estimación de costos de proyectos desde la  
Plataforma Cubana de Migración a Código  
Abierto**

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

**Autores**

Lisandra Graelles Fardales, Rogelio Padrón Caballero

**Tutores**

Ing. Abel García Vitier, Ing. Yasiel Pérez Villazón

**Consultante**

MBA. Ihoandra Sotolongo Carballo

*La Habana, Cuba, Junio 2015.  
"Año 57 de la Revolución"*

## **Declaración de autoría**

Declaramos ser los únicos autores de este trabajo y reconocemos a la Universidad de las 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 2015.

---

Lisandra Graelles Fardales

---

Rogelio Padrón Caballero

---

Ing. Abel García Vitier

---

Ing. Yasiel Pérez Villazón

## Agradecimientos

*Lisandra Graelles Fardales:*

*A mis padres, a mi hermana por su constante apoyo y estar presentes en todo momento y a mi tía Tamara por ser una segunda madre.*

*A mi compañero de tesis por formar parte de este momento y de mi vida en general, y por todo el apoyo brindado durante el desarrollo de este trabajo.*

*A nuestros tutores, por guiarnos siempre y de la mejor manera posible.*

*A nuestra consultante Ihoandra Sotolongo Carballo por ayudarnos en los temas asociados a su especialidad e incluso más.*

*A los trabajadores del departamento SIMAYS por la ayuda brindada durante la investigación, en especial nuestra oponente Susana Sánchez Ortiz, el director del centro CESOL Yoandy Pérez Villazón y el especialista de migración Jailen García González.*

*A mis compañeros de aula por contribuir a mi formación profesional y en especial a Laura Pérez Vera por estar presente durante los cinco años de la carrera y Marycarmen Díaz Labrador por brindarme su amistad.*

*Rogelio Padrón Caballero*

*Agradezco de todo corazón y de manera muy especial a mis padres, sin los cuales no hubiese sido posible finalizar mis estudios. A ellos que tanto esfuerzo han realizado durante toda mi carrera para que pudiera llegar hasta este momento en que se ve el fruto de tanto sacrificio y dedicación.*

*A mis tíos Icha y Tini que me cuidaron y me atendieron durante todo el tiempo en que mis padres estuvieron lejos de mí.*

*A Mayi por ayudarme en cada momento en que la he necesitado y atenderme durante todo este tiempo como si fuera otro hijo para ella.*

*A mi compañera de tesis por formar parte de este momento y de mi vida en general.*

*A mis tutores por todo el apoyo recibido, sus críticas, sus recomendaciones y por su confianza en la realización de este trabajo.*

*A mis compañeros de aula, a todos los que han compartido conmigo desde primer año hasta quinto, especialmente a aquellos que me apoyaron en los momentos difíciles: Ronaldo Castellanos que siempre me acompañó en las buenas y en las malas y Carlos Manuel Ferras por ayudarme en mi preparación profesional.*

## **Dedicatoria**

*Lisandra Graelles Fardales:*

*A mis padres y mi tía Tamara, quienes han soñado este momento conmigo y con el mismo anhelo.*

*Rogelio Padrón Caballero:*

*A toda mi familia por el apoyo incondicional que me ha brindado siempre y en especial a mis padres que han sido mi mayor motivación durante la carrera.*

## Resumen

En Cuba se han definido un conjunto de buenas prácticas que permitan planear, gestionar y ejecutar proyectos de migración a código abierto. Para automatizar algunas de las actividades comprendidas en dichas prácticas fue desarrollada en la Universidad de Ciencias Informáticas (UCI) una primera versión de la Plataforma Cubana de Migración a Código Abierto (PCMCA). Sin embargo, aún los análisis económicos se realizan manualmente. Esta investigación tiene como objetivo desarrollar funcionalidades para la generación de fichas de costo en el subsistema de Planificación, Control y Seguimiento (PCS) de la PCMCA que permitan una eficiente estimación de costos de los proyectos de migración a código abierto. A partir del análisis de sistemas de gestión de costos como Odoo, OpenBravo y Xedro GESPRO, la definición de requisitos funcionales por el cliente y el estudio de la propia arquitectura de la PCMCA, se detalla el proceso de desarrollo de las nuevas funcionalidades usando las prácticas que establece la metodología ágil SXP. Finalmente se validan los resultados a partir de pruebas de aceptación aplicadas al *software* y la técnica de V. A. Iadov para medir el nivel de satisfacción de los especialistas de migración. De esta manera se agrega valor a la PCMCA, permitiendo que pueda posicionarse entre los productos líderes de su tipo y contribuyendo a aportar mayor eficiencia a los procesos de migración que se gestionen con el uso de este sistema.

**Palabras claves:** *migración, código abierto, plataforma, estimación de costos, costos, ficha de costos.*

## Índice

Introducción.....	1
Capítulo 1: Marco teórico asociado a la estimación de costos de un proyecto de migración a código abierto.....	8
1.1:Principales conceptos.....	8
1.2:Sistemas informáticos libres o de código abierto que realizan gestión de costos.....	11
1.3:Descripción del proceso de migración a código abierto.....	13
1.4:Proceso de migración desde la PCMCA.....	16
1.5:Proceso de estimación de costos en un proyecto de migración a código abierto.....	17
1.6:Descripción del subsistema PCS.....	19
1.7:Descripción de la arquitectura de la PCMCA con énfasis en el subsistema PCS.....	20
Capítulo 2: Diseño e implementación de las funcionalidades para la estimación de costos de proyectos de migración a código abierto.....	22
2.1:Propuesta de solución.....	22
2.2:Lista de requisitos para la estimación de costos desde la PCMCA.....	24
2.1:Historias de usuario.....	29
2.2:Tecnologías y herramientas para el diseño e implementación.....	31
2.3:Modelo de diseño.....	33
2.3.1:Clase contactoActions.....	37
2.3.2:Clase costosIndirectosActions.....	39
2.3.3:Clase departamentoActions.....	40
2.3.4:Clase dependenciaActions.....	41
2.3.5:Clase faseActions.....	42
2.3.6:Clase fichaCostoActions.....	42
2.3.7:Clase miembroActions.....	43
2.3.8:Clase proyectoActions.....	44
2.3.9:Class recursoActions.....	48
2.3.10:Class tareaActions.....	49
2.3.11:Class tipoRecursoActions.....	53
2.4:Estructura de componentes.....	54
Capítulo 3: Validación de la solución propuesta.....	57
3.1:Casos de Prueba de Aceptación.....	57
3.2:Técnica de V. A. Iadov.....	58

3.3:Operacionalización de las variables.....	61
Conclusiones.....	64
Recomendaciones.....	65
Referencias bibliográficas.....	66
Bibliografía.....	69
Anexos.....	73



## **Introducción**

La dirección de proyectos es la aplicación de conocimientos, habilidades, herramientas y técnicas a las actividades del proyecto para cumplir con sus requisitos. Se logra mediante la aplicación e integración adecuadas de los procesos de la dirección de proyectos, agrupados de manera lógica, categorizados en cinco Grupos de Procesos: *Inicio, Planificación, Ejecución, Monitoreo y control, y Cierre* (PMI, 2013).

Existe una premisa que estipula que la ejecución de un proceso o proyecto implica la realización de un conjunto de actividades, utilizando el esfuerzo de recursos humanos, con tiempos de ejecución y costos a evaluar, entre otras variables que se estimen asociadas al mismo. En los casos que esos proyectos o procesos se ejecuten con alta demanda tecnológica y equipos de cómputo, habrán asociados altos costos por concepto de adquisición, actualización y soporte de tecnologías, en su mayoría privativas y con licenciamiento que también reviste un costo adicional relevante.

Para determinar y dar seguimiento a dichos costos, se realiza la Gestión de los Costos del Proyecto, el cual incluye los procesos relacionados con planificar, estimar, presupuestar, financiar, obtener financiamiento, gestionar y controlar los costos de modo que se complete el proyecto dentro del presupuesto aprobado (PMI, 2013). Los datos que la gestión de costos pueda ofrecer son de gran influencia en la toma de decisiones de una empresa, motivo por el cual se hace necesaria la veracidad y exactitud de los mismos. Las personas que manipulan los datos de entrada para el proceso deben estar bien capacitadas, así como el lugar donde se registra la información debe tener las condiciones necesarias para evitar su pérdida o modificación.

Debido a lo planteado anteriormente, muchas empresas han identificado la necesidad de automatizar la estimación de costos integrándolo a sistemas de gestión de proyectos, lo cual permite realizar el proceso en menor tiempo, con menos personal y tener un mayor control sobre la información que se introduce al

sistema y los resultados a obtener. Ello trae consigo además la reducción del costo del proyecto en general.

El auge de las tecnologías de código abierto ha demostrado que con la gradual incorporación de las instituciones públicas al uso de dichas tecnologías indudablemente los costos se reducen, a la vez que los sistemas o aplicaciones de código abierto son en su mayoría gratis y su código fuente puede ser adquirido y adaptado a las necesidades de las empresas. Por supuesto, siempre se realiza una proyección de los objetivos o metas trazadas con la ejecución del proceso y se establecen planes de migración a las nuevas tecnologías con sus respectivas especificaciones detalladas.

En los planes de migración quedarán reflejadas por consiguiente las actividades a realizar y sus variables de costos asociadas para posteriormente realizar un análisis del ciclo de vida del proyecto, en el cual intervengan las variables y sus relaciones y permita fijar criterios para decidir la ejecución o no del proyecto.

A partir de la necesidad de automatizar muchas de las actividades realizadas en un proceso de migración a código abierto, se comenzó a desarrollar la PCMCA, que entre sus beneficios incluye la optimización en los tiempos de ejecución de algunas actividades, la no duplicidad de información y la gestión de conocimiento (Diosbel Pérez et al. 2014).

Esta plataforma integra un conjunto de subsistemas entre los que se encuentra PCS que se encarga del control de proyectos de migración y toda su información relativa a entidades, tareas, recursos y riesgos. Permite además, planificar el proceso de migración y controlar su avance a partir de indicadores. El subsistema contiene los siguientes módulos:

- **Proyectos de migración:** se gestionan los proyectos de migración permitiendo definir alcance, presupuesto, impacto, entre otros indicadores.

- Modalidades, Prioridades, Fases y Flujos: estos módulos permiten configurar la organización de un proyecto de migración.
- Tareas: se gestionan las tareas con el fin de darles cumplimiento en tiempo.
- Roles y Recursos: se gestionan los actores que participan en los procesos de migración y los recursos materiales que se necesiten.
- Riesgos: se gestionan los riesgos de los proyectos de migración así como las acciones para su mitigación.

Si bien es cierto que la PCMCA contribuye a la automatización de tareas asociadas a la gestión de los proyectos de migración a código abierto y que con el uso del subsistema PCS se pueden gestionar las actividades y recursos involucrados, aún los análisis económicos se realizan manualmente. Esto por consiguiente atenta contra una gestión de costos eficiente por las siguientes razones:

- El proyecto se gestiona en la PCMCA y los costos en hojas de cálculo, por lo que se duplica información relacionada con las actividades y recursos involucrados en el proyecto.
- Existe mayor riesgo de cometer errores, sobre todo cuando hay cambios en los datos asociados al proyecto y deben ajustarse, tanto en la PCMCA, como en la ficha de costos; así como al manejar varias versiones de un mismo documento.
- No se centraliza la información referida a los costos del proyecto, pues estas hojas de cálculo se comparten vía correo electrónico con los especialistas que participan en el proceso de estimación. Esto incide directamente en una acertada toma de decisiones.

A las razones mencionadas puede agregarse que es vital para dotar efectividad al proceso de migración, aprender de experiencias exitosas y aplicar buenas prácticas. Es por ello que deben crearse las

condiciones que permitan la gestión del conocimiento generado durante el proceso de estimación de costos. Por lo que se necesita la creación y utilización de proyectos tipo, pudiendo entonces crear nuevos proyectos a partir otros con características semejantes, trayendo consigo ahorro de tiempo y esfuerzo al equipo de proyecto.

Teniendo en cuenta lo anterior, se plantea como **problema a resolver**: ¿cómo estimar eficientemente los costos de un proyecto de migración a código abierto?

El **objeto de estudio** lo constituye, por tanto, el proceso de estimación de costos de los proyectos de migración a código abierto, siendo el subsistema PCS de la PCMCA el **campo de acción** de la presente investigación.

Para dar solución al problema identificado se define como **objetivo general** desarrollar funcionalidades para la generación de fichas de costo en el subsistema PCS de la PCMCA que permitan una eficiente estimación de costos de los proyectos de migración a código abierto, el cual se desglosa en los siguientes **objetivos específicos**:

- Caracterizar el proceso de estimación de costos de proyectos de migración a código abierto.
- Diseñar e implementar las funcionalidades identificadas para la estimación de costos de proyectos de migración a código abierto en el subsistema PCS.
- Validar la solución propuesta.

Para dar cumplimiento a los objetivos específicos se formularon las siguientes **tareas**:

- Descripción de las actividades a realizar para estimar costos en un proyecto de migración a código abierto.
- Análisis de los sistemas informáticos libres o de código abierto que realizan estimación de costos.

- Descripción de la arquitectura de la PCMCA con énfasis en el subsistema PCS.
- Definición de los requisitos funcionales y no funcionales.
- Diseño e implementación de las clases y métodos que brindan solución a los requisitos definidos.
- Diseño y ejecución de casos de prueba a las funcionalidades implementadas.
- Medición del grado de optimización del tiempo empleado para estimar costos usando la solución desarrollada.

Se plantea como **hipótesis** que si se desarrollan funcionalidades para la generación de fichas de costo en el subsistema PCS de la PCMCA se contribuirá a una eficiente estimación de costos de los proyectos de migración a código abierto. Se define como variable independiente las funcionalidades para generación de fichas de costo y como variable dependiente la eficiente estimación de costos de los proyectos de migración a código abierto. En la Tabla 1 se detalla la operacionalización de las variables.

Tabla 1: Operacionalización de las variables de la hipótesis.

Tipo de variable	Variable	Dimensión	Indicador	Unidad de medida
Independiente	Funcionalidades para generación de fichas de costo	Satisfacción de los especialistas de migración.	Grado de satisfacción	Satisfecho, no satisfecho, contradictorio
		Cumplimiento de los requisitos.	Resultado de las pruebas de aceptación.	Satisfactoria, no satisfactoria.
Dependiente	Eficiente estimación de costos de los proyectos de	Tiempo	Duración del proceso de estimación de costos.	Segundos
		Recursos	Cantidad de recursos humanos necesarios para el proceso de estimación de	Unidades

	migración a código abierto		costos.	
			Cantidad de recursos materiales necesarios para el proceso de estimación de costos.	Unidades

A los efectos de esta investigación eficiencia está orientada a los medios utilizados para alcanzar los objetivos. La eficiencia es alcanzada cuando se logran los mismos resultados utilizando menos recursos, o bien cuando, utilizando los mismo recursos se consiguen mejores resultados (Juan et al. 2014).

Durante el desarrollo de la investigación se emplearon los siguientes **métodos científicos**:

### **Métodos Teóricos**

Analítico - Sintético: se empleó en la descomposición del proceso de estimación de costos en varios subprocesos e identificando en cada uno los diferentes aspectos de interés, así como el flujo entre ellos y su correspondencia con las funcionalidades y datos manejados en el subsistema PCS.

Hipotético - Deductivo: a partir de los conocimientos sobre la estimación de costos, permitió predecir las mejoras que traerá consigo su automatización, asociado a los proyectos de migración a código abierto manejados en la PCMCA.

Método de modelación: permitió representar los procesos y actores involucrados en la estimación de costos, así como los componentes que intervienen en la implementación de la solución y las nuevas clases incorporadas.

### **Método Empírico**

Método de Observación: permitió medir los indicadores de la variable eficiencia mediante su comportamiento durante el proceso de estimación de costos realizado tanto manualmente como desde la PCMCA.

### **Métodos Particulares**

Entrevista: permitió recopilar información a través de personal capacitado en la gestión y estimación de costos para definir las variables y actividades a tener en cuenta durante la investigación.

Encuesta: se empleó para conocer el nivel de satisfacción de los especialistas de migración respecto a las funcionalidades implementadas.

Este documento está compuesto por una introducción, tres capítulos, conclusiones generales, referencias bibliográficas, bibliografía y el glosario de términos donde se explican los vocablos de difícil comprensión que se han empleado en el documento. La estructura de los capítulos se define a continuación:

**Capítulo 1:** Marco teórico asociado a sistemas o aplicaciones que realizan la gestión de costos y su caracterización en los proyectos de migración a código abierto llevados a cabo en la universidad con el fin de identificar las funcionalidades necesarias a incluir en el subsistema PCS. Para ello se enuncian los diferentes conceptos asociados a la gestión de costos y los proyectos de migración a código abierto.

**Capítulo 2:** Descripción de los principales artefactos generados en el proceso de desarrollo de la solución propuesta, según la metodología empleada.

**Capítulo 3:** Validación de la solución haciendo uso de pruebas de aceptación aplicadas por el cliente y la técnica de V. A. Iadov para medir el nivel de satisfacción de los especialistas de migración con los resultados obtenidos.

## **Capítulo 1: Marco teórico asociado a la estimación de costos de un proyecto de migración a código abierto**

En el presente capítulo se abarcan los principales conceptos asociados al objeto de estudio, permitiendo una mejor comprensión del tema. Se analizan sistemas informáticos existentes que manejan la gestión de costos; estudiando todas las vías posibles para realizar la estimación de costos de un proyecto de migración a código abierto. También se describe el proceso de migración con sus principales tareas y la estimación de costos enfocado en este. Finalmente se hace alusión a la arquitectura presente en la PCMCA y por consiguiente en el subsistema PCS.

### **1.1: Principales conceptos**

Dado el vínculo de esta investigación con las ramas de la Contabilidad y la Economía se hace necesario mencionar algunos conceptos para una mejor comprensión, comenzando con la definición de costo por ser la base para los demás conceptos.

**Costo:** gasto económico que representa la fabricación de un producto o la prestación de un servicio (Jiménez 2014).

**Costo total:** Es la suma de costos de las personas que realizan el trabajo, el costo de los recursos empleados en el mismo y costos varios de soporte al proyecto (Curto, 2013).

Al asignar un recurso a una actividad, este puede estar directamente vinculado a su desarrollo, o estar asociado a varias tareas a la vez, como es el caso de los recursos vinculados al proyecto en general e inciden de forma indirecta en las actividades. Los costos relacionados con cada recurso se manejan de forma diferenciada, para ello se clasifican en:

**Costos directos:** están directamente relacionados al aplicativo resultante y pueden ser reconocidos en este mediante un fácil mecanismo de seguimiento efectivo en términos de costo (López, 2013). En estos



se incluyen el costo de salarios y prestaciones de los actores directos, en este caso, los especialistas de migración y el costo incurrido por el uso de los recursos materiales necesarios para la ejecución de las tareas.

**Costos indirectos:** están relacionados con el producto final pero no pueden ser identificados en este mediante una simple formula económica (López, 2013). En este caso se incluyen los salarios y prestaciones de los actores indirectos como pueden ser los encargados de la administración de los proyectos.

Los costos anteriormente mencionados se obtienen a partir de la **estimación de costos:** proceso que consiste en desarrollar una aproximación de los recursos monetarios necesarios para completar las actividades del proyecto (PMI, 2013).

La estimación de costos según el nivel de detalle, exactitud y uso que se pretenda darle, se clasifica en:

1. Estimaciones del orden de magnitud: se usan en la etapa de planeación y evaluación inicial de un proyecto (Sullivan, Wicks y Luxhoj, 2004).
2. Estimaciones de semidetalle o presupuesto: se emplean en la etapa de diseño preliminar o conceptual de un proyecto (Sullivan, Wicks y Luxhoj, 2004).
3. Estimaciones definitivas: se utilizan en la etapa de ingeniería de detalle o construcción de un proyecto (Sullivan, Wicks y Luxhoj, 2004).

La estimación de costos concerniente a la investigación es la de orden de magnitud, teniendo en cuenta la etapa del proyecto en la que se realiza. Para llevar a cabo la estimación, son aplicables varias técnicas, entre las que se encuentran:

1. Índices: un índice es un número adimensional que indica cómo ha cambiado con el tiempo un costo con respecto a un año base. Los índices proporcionan un medio conveniente para desarrollar

estimaciones presentes y futuras del costo a partir de datos históricos (Sullivan, Wicks y Luxhoj, 2004).

2. Técnica unitaria: implica el uso de un factor por unidad como pueden ser: el costo de construcción por pie cuadrado o el costo de mantenimiento por hora. Tales factores multiplicados por la unidad apropiada, arrojan una estimación del costo total (Sullivan, Wicks y Luxhoj, 2004).
3. Técnica del factor: es una extensión del método unitario, dentro de una estrategia básica de segmentación, en la que se suma el producto de varias cantidades o componentes y se agrega a cualesquiera componentes estimados en forma directa (Sullivan, Wicks y Luxhoj, 2004).

Al tratarse de proyectos de migración donde el costo total de una actividad se calcula a partir de las horas dedicadas para su ejecución por parte de los recursos tanto materiales como humanos, la depreciación de los materiales se debe calcular basándose en dichas horas, unidad de medida empleada en cada cálculo que se realice en el proceso de estimación. Por tal motivo se decide escoger la **técnica unitaria** para dicho proceso.

Existen recursos materiales cuyo tiempo de vida es prolongado, por lo que pueden ser usados en más de un proyecto. Esto hace que cada vez que sean asignados a una actividad, pierdan valor debido al uso y desgaste que van adquiriendo. Este fenómeno se define como:

**Depreciación:** término contable que denota la disminución en el valor de un activo fijo tangible debido al deterioro físico o desgaste natural donde el propio tiempo desgasta gradualmente un bien de capital, independientemente de que se utilice o no (Jiménez, 2014).

Partiendo del concepto anterior se debe tener en cuenta que cada recurso material tiene coligado una tasa de depreciación que permite calcular el valor que deprecia por cada hora de uso. Para realizar el cálculo existen varios métodos:

1. Método de la línea recta (LR): como su nombre lo indica, el método de depreciación en línea recta asigna una cantidad igual del costo de un activo para el gasto de depreciación por cada año de la vida útil del activo (Rich et al. 2013).
2. Método del saldo decreciente (SD): es un método de depreciación acelerada que produce una cantidad cada vez menor de los gastos de depreciación de cada período, multiplicando el valor contable decreciente de un activo por una tasa de depreciación constante (Rich et al. 2013).
3. Método de las unidades de producción: se utiliza cuando la disminución del potencial de servicio de un activo es proporcional a su uso y este puede ser medido. Normalmente se mide por la capacidad productiva como las unidades producidas, las horas trabajadas o millas conducidas (Rich et al. 2013).

De los anteriores se emplea el método de **Línea Recta** al ser el más usado por su simplicidad y facilidad de aplicación (Rich et al. 2013).

Reunidos los datos y realizada la estimación, se elabora una declaración de los costos incurridos, o por incurrir, para la producción de un determinado volumen de producción o de prestación de servicios, siendo este último el caso a tratar. Dicha declaración es denominada **ficha de costos** (Banerjee, 2009).

Conocidos los principales conceptos involucrados en el proceso de estimación de costos se pueden identificar con más exactitud aquellos sistemas informáticos que realicen la gestión y por consiguiente la estimación de costos.

### **1.2: Sistemas informáticos libres o de código abierto que realizan gestión de costos**

La gestión de proyectos es uno de los procesos cuya automatización ha promovido en muchas empresas y comunidades la creación de sistemas informáticos, tanto privativos como de *software* libre y código abierto. A los efectos de la presente investigación y con el objetivo de encontrar soluciones ya implementadas que permitan realizar de forma automática la estimación de costos, se decide escoger

para su estudio sistemas informáticos de código abierto, que realicen la gestión de proyectos y a su vez la gestión de costos.

Las soluciones ERP (del inglés, *Enterprise Resource Planning*) están orientadas a facilitar la integración de los sistemas de las empresas, asegurar la comunicación y mejorar su productividad industrial o la eficiencia de la gestión financiera (Centro de Excelencia en Comercio Electrónico, 2011). Teniendo en cuenta los aspectos mencionados anteriormente se decide estudiar dos soluciones líderes de código abierto en el mercado ERP: OpenBravo y Odoo (anteriormente llamado OpenERP) (Sánchez, 2015).

### **Openbravo**

La plataforma de Openbravo es una solución de comercio construida sobre una plataforma de desarrollo web, móvil y lista para la nube, que incluye funcionalidad ERP horizontal. Esta permite diversas funcionalidades, entre las que se encuentra la gestión corporativa, la cual cuenta con una gestión financiera que incluye contabilidad, cuentas a pagar y cobrar, y gestión de activos. Una funcionalidad multimoneda, multimpuestos y multicontabilidad que proporciona a los minoristas el soporte necesario para sus operaciones internacionales, incluyendo informes financieros y de impuestos de gran calidad para cumplir las distintas normativas legales y fiscales (Openbravo, 2014).

La información que se maneja en la plataforma de Openbravo está relacionada mayormente con el comercio electrónico, enfocada en los productos en venta, estrategias de marketing, clientes, proveedores, entre otros. Partiendo de lo anterior se plantea que los costos que se gestionan están asociados a los servicios que se brindan en internet como el trato de las transacciones y cuentas bancarias de los clientes.

### **Odoo**

Solución ERP de código abierto que integra funciones para ventas, CRM, gestión de proyectos, contabilidad, gestión de almacenes e inventarios, fabricación, gestión financiera, tienda *online* y recursos

humanos, entre otros. Además, permite trabajar remotamente mediante una interfaz web desde un ordenador o dispositivo móvil conectado a Internet (Centro de Excelencia en Comercio Electrónico, 2011).

Los datos manejados en el módulo de contabilidad no están relacionados con los proyectos creados en el sistema, sino con la empresa en general. Son datos de administración y contabilidad de la empresa, sin llegar a vincularse directamente con los proyectos que en esta se desarrollen.

### **Xedro GESPRO**

Paquete para la toma de decisiones y la dirección integrada de proyectos. Está formado por dos elementos fundamentales: un sistema informático GESPRO y un paquete de servicios para la formación de competencias profesionales en gestión de proyectos. Entre sus módulos se encuentra Gestión de costos y facturaciones, el cual permite el manejo de los presupuestos y costos de los proyectos con facilidades para la integración con sistemas ERP. Incluye posibilidades para el seguimiento de las facturaciones y la gestión del flujo de caja (Pérez, 2013).

El sistema permite asociar costos a los hitos de los proyectos y no a las tareas. Si se crea una tarea asociada a un hito y esta se modifica, los costos no se actualizan automáticamente, dificultando así el trabajo del usuario al tener que realizarlo de forma manual. Al crear un recurso, uno de los parámetros que necesita ser llenado es *Tasa de costo*, el cual representa para los recursos humanos el valor hora-hombre y para los recursos materiales la depreciación en hora. En el caso de los recursos materiales, el tener que insertar este valor, trae consigo tener que calcularlo anteriormente implicando trabajo manual. Como aspecto positivo se encuentra el poder mostrar un resumen de los costos del proyecto y gráficas que muestran información a partir del análisis de costos.

Teniendo en cuenta los aspectos mencionados anteriormente de cada uno de los sistemas informáticos, se concluye que ninguno brinda solución al problema planteado en la investigación, por lo que se decide

crear nuevas funcionalidades en la PCMCA que permitan realizar el proceso de estimación de costos de manera automática.

### **1.3: Descripción del proceso de migración a código abierto**

Crear nuevas funcionalidades en la PCMCA implica conocer las actividades que en ella se realizan, lo que conlleva a describir los procesos que se llevan a cabo en un proyecto de migración. Para ello primeramente se define la migración a tecnologías de *software* libre y código abierto.

Una migración a tecnologías de *software* libre y código abierto puede definirse como: *“Un proceso ordenado en el cual se sustituye, parcial o totalmente, el software existente en la organización por alternativas liberadas bajo licencias libres o de código abierto. Es recomendado incluir la adopción de estándares abiertos para la documentación”* (Villazón et al. 2014).

Como proceso ordenado, la migración debe contar con personas encargadas de guiar cada paso y actividad que se desarrolle, según sus habilidades y conocimientos hasta lograr el éxito del proceso. Estas personas deberán representar los roles (Villazón et al. 2014):

1. Jefe del equipo de migración.
2. Especialista de capacitación.
3. Especialista de soporte técnico.
4. Especialista de relaciones humanas.
5. Especialista en servicios telemáticos.
6. Especialista en migración.

En dependencia de la complejidad del proyecto, teniendo en cuenta el tamaño de la empresa y las habilidades del equipo de migración, una misma persona puede desempeñar más de un rol.

El éxito de una empresa no depende únicamente de la tecnología, sino que las personas juegan un papel importante en su desarrollo. Durante un proceso de migración hay que tener en cuenta ambas partes, ya que migrada la empresa, sin un personal capacitado y satisfecho con los resultados, puede provocar el rechazo a los cambios tecnológicos y finalmente el fracaso del proceso. Debido a esto se establece llevar a cabo la migración social y la migración tecnológica simultáneamente. Para ello están definidas actividades de capacitación y un orden estratégico de las actividades que permite a los usuarios familiarizarse poco a poco con el nuevo sistema (Villazón et al. 2014).

Un proyecto de migración está compuesto por tres etapas: Preparación, Ejecución y Consolidación (Villazón et al. 2014).

La etapa de *Preparación* es donde se realiza el diagnóstico de *hardware* y *software*, y un estudio del negocio y los procesos de la institución. Lo que permite obtener el inventario de todas las tecnologías, desde las máquinas de escritorio, hasta los servidores y servicios telemáticos, así como los dispositivos externos. También se tienen en cuenta aquellos empleados identificados como líderes de opinión, con quienes se hará un trabajo más riguroso en cuanto a la opinión sobre el *software* libre, debido a que son estos quienes más influyen en el éxito o no de la migración. Esta etapa termina con el diseño de una hoja de ruta a seguir por la institución a migrar para lograr los cambios que exige el proceso.

Teniendo toda la información generada en la *Preparación*, comienza la etapa de *Ejecución*. Primeramente se migran los servicios telemáticos, luego de haber capacitado al personal de redes. Durante la actividad se tendrán en cuenta las copias de seguridad a los sistemas, para posteriormente ejecutar la migración. Una vez concluido este paso, se finaliza con la definición de las medidas de seguridad incluyendo la política de salvadas. Llegado a este punto, están las condiciones creadas para migrar el resto de la institución. Luego de informar a los trabajadores sobre el procedimiento, se instalan herramientas libres sobre el propio sistema privativo, se imparten los cursos de capacitación necesarios y

posteriormente, se migran los archivos generados con las herramientas privativas haciéndolas compatibles con las de código abierto. Finalmente se procede a cambiar el sistema operativo base.

El soporte debe ponerse en funcionamiento luego de planificar la migración pero antes de ejecutarla. Esto permite que mientras se vayan realizando los cambios en la empresa, se pueda dar solución a los problemas que vayan surgiendo, aprovechando el tiempo y los recursos al máximo, y evitar que aparezcan nuevamente en el resto de las actividades. No obstante, ya concluida la migración, se continúa dando soporte durante el tiempo pactado. Para comenzar se evalúa el punto de partida existente, donde se recogen las condiciones actuales de la empresa. Luego se define un protocolo de atención a las incidencias, que determinará el orden de las incidencias para ser atendidas, según el impacto de cada una. También es necesario definir un mecanismo para la gestión de incidencias y los mecanismos tecnológicos de apoyo, que son los que permitirán guiar los pasos para su solución. Reunida toda la información, se elabora finalmente el plan de soporte, para dar inicio a las actividades (Ortiz, Benitez y Hernández, 2015). Los pasos mencionados anteriormente conforman el proceso a realizar en la etapa de *Consolidación*.

Para la realización del proceso de migración son necesarios determinados recursos, entre los que se encuentra un disco duro donde se almacenan los repositorios y la documentación de los cursos de capacitación. Por otra parte son necesarias las computadoras, impresoras y escritorios que son usados en las actividades de planificación, entre otras. Estos recursos son clasificados como equipos de cómputo y muebles, respectivamente. En la etapa de *Ejecución* están presentes materiales de oficina para las capacitaciones e insumos informáticos, que no son más que los DVD que contienen la imagen del nuevo sistema operativo a instalar.

#### **1.4: Proceso de migración desde la PCMCA**

La PCMCA cuenta con varios subsistemas que permiten gestionar toda la información que se genera en un proyecto de migración. Los subsistemas PCS, Seguridad y Directorio de *software* fueron liberados en



**Capítulo 1: Marco teórico asociado a la estimación de costos de un proyecto de migración a código abierto**

---

---

su primera versión, sin embargo actualmente se encuentran en desarrollo Migración, Inventario, Certificaciones y Encuestas. El subsistema PCS interviene durante todo el proceso de migración, planificando y controlando cada una de las actividades, además de tener registrada toda la información asociada a los proyectos. Por otra parte Migración es donde se reúnen todos los datos generados en los demás subsistemas y se manejan cada una de las actividades que se realizan en la etapa de Ejecución. El resto de los subsistemas están presentes en la etapa de Preparación durante los diagnósticos de *hardware* y *software*, y las encuestas realizadas al personal de la entidad involucrada, Figura 1.

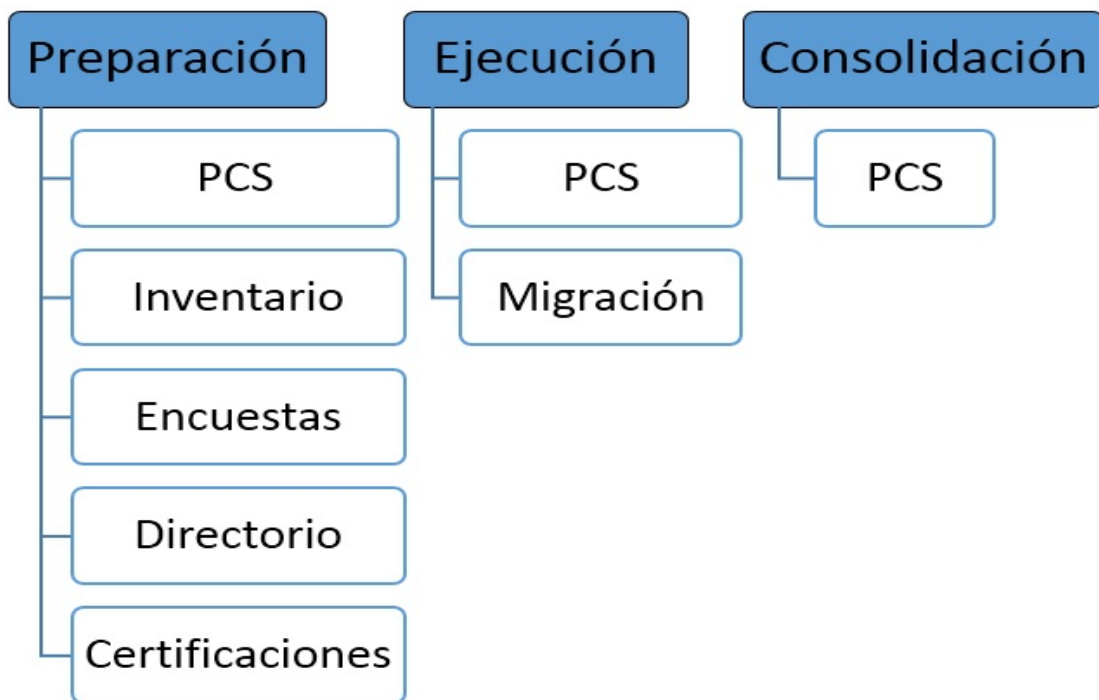


Figura 1: Subsistemas asociados al proceso de migración.

### **1.5: Proceso de estimación de costos en un proyecto de migración a código abierto**

Con el objetivo de poder realizar el proceso de estimación de costos desde la PCMCA, es necesario conocer primeramente los datos que intervienen en el proceso, así como las actividades que lo conforman.

Recibida la solicitud formal de una institución para migrar, se determina el alcance del proyecto, teniendo en cuenta el tipo de migración que se solicita (parcial o completa) y el tamaño de la empresa según la cantidad de computadoras que posee. A continuación se planifican las actividades y se asignan los recursos Figura 2. Conociendo las tasas de depreciación de cada recurso material y el valor hora-hombre de los recursos humanos asignados, además de la duración de cada actividad planificada, se tienen los datos necesarios para estimar mediante juicio de expertos, los costos totales del proyecto Figura 3.

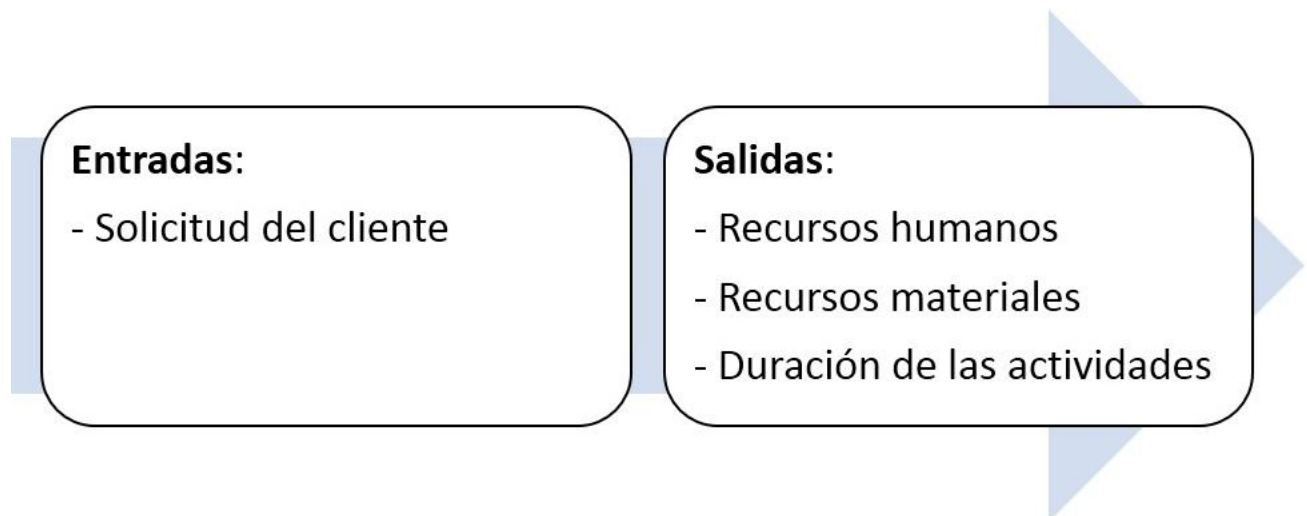


Figura 2: Planificación de las actividades de un proyecto.

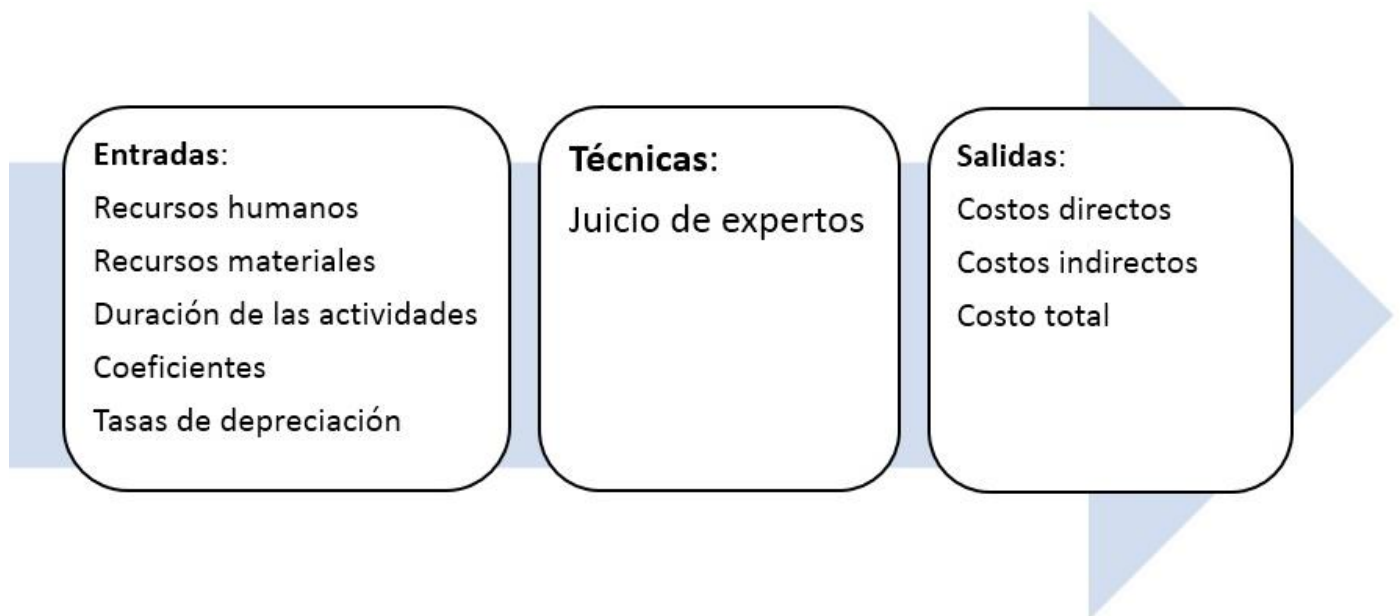


Figura 3: Estimación de costos de un proyecto de migración a código abierto.

### **1.6: Descripción del subsistema PCS**

Siendo PCS el subsistema donde se gestionan los proyectos de migración y por tanto donde se manejan los datos necesarios para la estimación de costos de un proyecto, es en dicho subsistema donde deben ser incorporadas las nuevas funcionalidades. Este contiene varios módulos, donde algunos de ellos tienen como función gestionar las configuraciones de los proyectos como son *Modalidades*, *Prioridades*, *Flujos* y *Fases*. Este último, al crear una nueva fase, no incluye la prioridad, lo cual trae conflictos a la hora de definir las fases de un proyecto y que estas queden ordenadas correctamente. En el caso de *Roles* y *Recursos*, se trata de registrar los recursos necesarios y disponibles para el equipo de proyecto, sin embargo, no existe vínculo entre la disponibilidad existente y los que ya han sido asignados, por lo que

dicha disponibilidad no se actualiza de forma automática imposibilitando un correcto control de los mismos. Por otro lado, los recursos materiales no tienen asociadas las tasas de depreciación, dificultando así el proceso de estimación de costos desde la plataforma, sucediendo lo mismo con los recursos humanos, tratándose en este caso del valor hora-hombre.

Los módulos *Tareas* y *Riesgos* gestionan las actividades a realizar y los posibles problemas que puedan ocurrir durante el desarrollo del proyecto respectivamente. En el caso de las tareas, no es posible asignarle los recursos necesarios para su ejecución, pues solo son asignados a nivel de proyecto, no se vinculan a las fases del mismo, no registran información sobre su estado de desarrollo y no permiten establecer dependencias con otras tareas, datos necesarios para la elaboración del diagrama de Gantt y otros procesos que no están implementados y que son de gran importancia para el proceso de migración. Finalmente, todo lo vinculado a cada uno de los proyectos, dígame entidad involucrada y los datos de interés, así como el equipo de proyecto asignado y los contactos principales de ambas partes son gestionados desde el módulo de *Proyectos de migración*. Este último tiene como dificultades que no brinda la posibilidad de crear proyectos tipo, los cuáles facilitarían la planificación siempre que se quiera crear un nuevo proyecto a partir de otro con características semejantes registrado con anterioridad. Por otra parte, al asignar los recursos humanos, no se tiene en cuenta el rol que este desempeña, lo cual impide llevar el control de los roles que aún no han sido cubiertos en el proyecto.

El estado actual del PCS no es satisfactorio, teniendo en cuenta las deficiencias existentes, agregando además que influyen directa e indirectamente en el proceso de estimación de costos, por lo que es necesario comenzar dando solución a cada una de ellas hasta lograr obtener los resultados esperados. Para ello es necesario conocer la estructura del subsistema en cuestión proporcionando claridad y conocimiento sobre su funcionamiento.

### **1.7: Descripción de la arquitectura de la PCMCA con énfasis en el subsistema PCS**

La arquitectura de *software* es una disciplina importante de la ingeniería de *software*. Como resultado del aumento en tamaño y complejidad de los sistemas informáticos, la especificación y diseño de la estructura general de tales sistemas se ha convertido en una parte cada vez más dominante en su desarrollo. La aplicación de la arquitectura más adecuada debería reducir el tiempo de desarrollo y aumentar la eficacia y la adecuación de la solución computacional resultante (Shehory y Sturm, 2014).

El subsistema PCS, al formar parte de la PCMCA tiene la necesidad de adaptarse a la arquitectura de la misma, que en este caso es Arquitectura Orientada a Servicios (SOA) y Presentación desacoplada (Vitier y González, 2011). El uso de SOA permite acceder a todas las funcionalidades de la PCMCA y de que todos sus componentes intercambien información mediante servicios web desde una interfaz. Por otro lado Presentación desacoplada simplifica el código de las interfaces para que la PCMCA pueda disponer de varias interfaces para una misma lógica de negocio.

## **Capítulo 2: Diseño e implementación de las funcionalidades para la estimación de costos de proyectos de migración a código abierto**

Conociendo las características de un proyecto de migración y su flujo manejado desde la PCMCA, además de los principales conceptos asociados al proceso de estimación de costos y algunos sistemas que realizan la gestión de costos, se procede a realizar el diseño e implementación de las funcionalidades a incorporar en el PCS. Para ello se describe la solución propuesta y se definen los requisitos funcionales y no funcionales, siendo los primeros, organizados por prioridad y posteriormente especificados a través de las historias de usuario (HU) con sus respectivas tareas de ingeniería. Se muestra además la estructura lógica y física del PCS.

### **2.1: Propuesta de solución**

Se propone implementar funcionalidades que permitan generar una ficha de costos para cada proyecto de migración a código abierto. Para obtener la ficha de costos son necesarios varios datos que deben estar registrados con anterioridad en el sistema, como son: la tasa de depreciación correspondiente a cada tipo de recurso material; el valor hora-hombre de los recursos humanos; la duración de las tareas y del proyecto en general dado por la planificación. La ficha de costos se genera nuevamente siempre que se ejecute la funcionalidad, permitiendo así mostrar cualquier variación de los valores a causa de algún cambio realizado. Las funcionalidades a implementar permiten llevar un control de tal forma que se muestren a través de gráficas las desviaciones del costo total del proyecto como consecuencia de los cambios realizados y el balance entre las partidas del costo total: costos directos, indirectos y gastos generales.

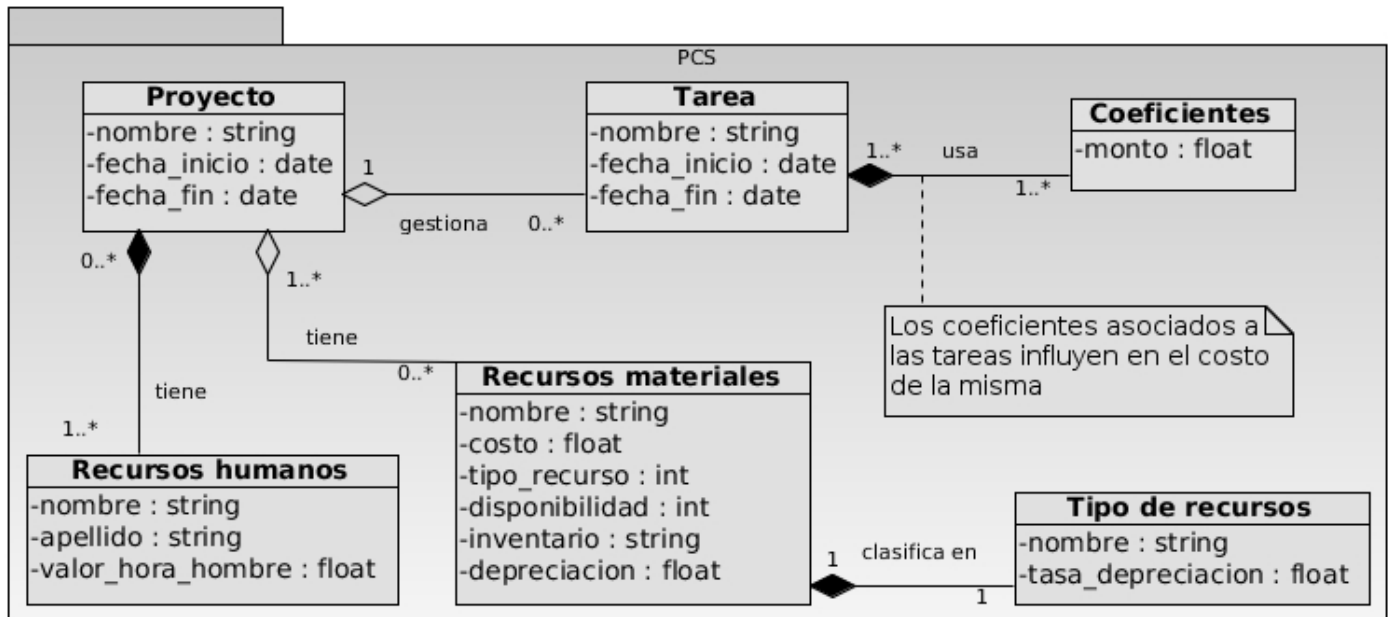


Figura 4: Propuesta de solución.

En la figura anterior se muestran las clases que intervienen en la estimación de costos de un proyecto y las relaciones entre ellas. Teniendo en cuenta las funcionalidades desarrolladas hasta el momento en el subsistema PCS y la solución propuesta, se hace necesario incluir la clase encargada de manejar los coeficientes y modificar el resto de las clases. Los recursos materiales son clasificados en tipos de recursos, quienes manejan la tasa de depreciación que da lugar a la depreciación de un recurso material. El proyecto es el responsable de los recursos tanto materiales como humanos y asignarlos a las tareas que en él se gestionan, al mismo tiempo que se encarga de generar la ficha de costos donde intervienen los coeficientes que fueron asignados a cada una de las tareas.

## **2.2: Lista de requisitos para la estimación de costos desde la PCMCA**

SXP es una metodología ágil de desarrollo que adopta las mejores prácticas de las metodologías SCRUM y XP (Romero, 2008), escogida para guiar el desarrollo de la PCMCA y por consiguiente la empleada en el subsistema PCS, convirtiéndose en la metodología que guía el desarrollo de la solución propuesta en la presente investigación comenzando por la propia definición de los requisitos.

Teniendo en cuenta la solución propuesta y los aspectos analizados en las reuniones con el cliente, fueron definidos los siguientes requisitos funcionales:

### **Requisitos funcionales**

#### **Prioridad alta:**

1. Listar tipos de recursos materiales.
2. Buscar tipo de recursos materiales.
3. Adicionar tipo de recursos materiales.
4. Editar tipo de recursos materiales.
5. Eliminar tipo de recursos materiales.
6. Listar recursos materiales disponibles para los proyectos.
7. Buscar recursos materiales disponibles para los proyectos.
8. Adicionar recursos materiales disponibles para los proyectos.
9. Editar recursos materiales disponibles para los proyectos.
10. Eliminar recursos materiales disponibles para los proyectos.



- 11.** Listar recursos humanos disponibles para los proyectos.
- 12.** Mostrar detalles de un recurso humano.
- 13.** Adicionar recursos humanos disponibles para los proyectos.
- 14.** Editar recursos humanos disponibles para los proyectos.
- 15.** Eliminar recursos humanos disponibles para los proyectos.
- 16.** Buscar tareas de un proyecto.
- 17.** Listar tareas de un proyecto.
- 18.** Mostrar detalles de una tarea.
- 19.** Adicionar tarea a un proyecto.
- 20.** Editar tarea de un proyecto.
- 21.** Eliminar tarea de un proyecto.
- 22.** Listar proyectos.
- 23.** Buscar proyecto.
- 24.** Mostrar detalles de un proyecto.
- 25.** Adicionar un proyecto.
- 26.** Editar un proyecto.
- 27.** Eliminar proyecto.

- 28. Mostrar recursos materiales de un proyecto.
- 29. Asignar recursos indirectos a un proyecto.
- 30. Mostrar proyectos de tipo referencia.
- 31. Mostrar detalles del proyecto de tipo referencia seleccionado.
- 32. Buscar coeficientes económicos.
- 33. Listar coeficientes económicos.
- 34. Adicionar coeficientes económicos.
- 35. Editar coeficientes económicos.
- 36. Eliminar coeficientes económicos.
- 37. Generar ficha de costos de un proyecto.
- 38. Exportar ficha de costos de un proyecto.

**Prioridad media:**

- 39. Mostrar tendencia del costo total de un proyecto.
- 40. Mostrar partidas del costo total de un proyecto.

**Prioridad baja:**

- 41. Listar fases de un proyecto.
- 42. Adicionar fase de un proyecto.

- 43. Editar fase de un proyecto.
- 44. Eliminar fase de un proyecto.
- 45. Adicionar entidad asociada a un proyecto.
- 46. Editar entidad asociada a un proyecto.
- 47. Mostrar detalles de entidad asociada a un proyecto.
- 48. Listar departamentos de una entidad.
- 49. Mostrar detalles de un departamento de una entidad.
- 50. Adicionar departamentos a una entidad.
- 51. Editar departamentos de una entidad.
- 52. Eliminar un departamento de una entidad.
- 53. Mostrar diagrama Gantt.
- 54. Mostrar detalles de un contacto de la entidad asociada a un proyecto.
- 55. Asignar contacto principal de la entidad asociada a un proyecto.

### **Requisitos no funcionales**

Al tratarse de funcionalidades incorporadas al PCS y este ser un subsistema de la PCMCA, los requisitos no funcionales que se enuncian a continuación corresponden a la propia PCMCA.

Usabilidad

- El sistema podrá ser usado sobre ambiente Web por personas con conocimientos relacionados con los procesos de migración guiados por la Metodología Cubana de Migración a Código Abierto.

#### Confiabilidad

- En caso de que el sistema presente alguna falla, los errores se deben mostrar sin detalles de información, ya que puede comprometer la seguridad e integridad del mismo.
- La información contenida en el sistema debe ser totalmente confiable.
- Se podrá acceder a la aplicación desde los navegadores web: Mozilla Firefox, Chrome y Opera.

#### Seguridad

- La información estará protegida contra accesos no autorizados utilizando mecanismos de validación que puedan garantizar el cumplimiento de esto: usuario, contraseña y nivel de acceso, de manera que cada uno pueda tener disponible solamente las opciones relacionadas con su actividad y tenga datos de acceso propios, garantizando así la confidencialidad.
- Se usarán mecanismos de encriptación de los datos que por cuestiones de seguridad no deben estar en texto plano, por lo que las contraseñas en la Base de Datos se almacenarán de forma encriptada.

#### Eficiencia

- Se esperan alrededor de 8 usuarios conectados concurrentemente.

#### Interfaz

- La interfaz de la plataforma debe estar acorde a las pautas de diseño establecidas por la UCI para la marca XILEMA

#### Interfaces de *hardware*

- Para explotación del cliente: PC Pentium 3 o superior, CPU 133 Mhz o superior; 256 RAM mínimo, recomendada 512 RAM o superior.
- Para el desarrollo: PC Intel Pentium 4 o superior, CPU 3GHz o superior, 2 GB de RAM o superior, 160 GB HDD o superior.

### **2.1: Historias de usuario**

Las historias de usuario (HU) son la técnica definida en la metodología SXP para especificar los requisitos del *software*, facilitando el proceso de estimación del tiempo de desarrollo y sirviendo como guía para la elaboración de las pruebas de aceptación. En el momento de implementar se deben detallar a través de la comunicación con el cliente.

Las tareas de ingeniería permiten definir cada una de las actividades que estarán asociadas a las HU dándole organización a su implementación. También posibilita conocer el tiempo que se necesita para su realización, lo que facilita la estimación del tiempo que se llevará cada historia de usuario en implementarse, de acuerdo a su complejidad (Romero, 2008).

La siguiente tabla recoge los principales aspectos de las HU. Para más detalles puede consultarse el Anexo 1.

Tabla 2: HU de la solución propuesta.

Nº	Nombre	Prioridad	Iteración	Descripción
HU 1	Gestionar recursos	Alta	1	Debe permitir listar, adicionar, editar, eliminar y

**Capítulo 2: Diseño e implementación de las funcionalidades para la estimación de costos de proyectos de migración a código abierto**

---



---

	humanos			mostrar detalles de recursos humanos del sistema.
HU 2	Mostrar recursos materiales de un proyecto	Alta	1	Se muestran los recursos materiales asignados al proyecto y la relación entre la disponibilidad y la cantidad asignada.
HU 3	Asignar recursos indirectos a un proyecto	Alta	1	Debe permitir asignar al proyecto recursos humanos y materiales que impliquen costos indirectos.
HU 4	Adicionar proyectos a partir de un proyecto de tipo referencia	Alta	1	Al crear un nuevo proyecto se debe poder seleccionar entre crear proyecto referencia, crear a partir de un proyecto referencia o ninguno de los anteriores.
HU 5	Editar coeficientes económicos	Alta	1	Debe permitir listar, adicionar, editar y eliminar los coeficientes registrados en el sistema.
HU 6	Generar ficha de costos de un proyecto	Alta	1	Se genera la ficha de costos de un proyecto, mostrando las partidas del costo total del mismo.
HU 7	Exportar ficha de costos	Alta	1	Permite exportar la ficha de costos de un proyecto en formato .csv.
HU 8	Mostrar las tendencias de los costos de un proyecto	Media	1	Se muestran dos gráficas que evidencian las desviaciones del costo de un proyecto junto al presupuesto real y las partidas del costo total.
HU 9	Mostrar diagrama de Gantt	Baja	2	Muestra el tiempo de dedicación previsto para las tareas del proyecto a lo largo de un tiempo total determinado. Representa las diferentes fases y tareas programadas así como las dependencias entre las mismas.

**Capítulo 2: Diseño e implementación de las funcionalidades para la estimación de costos de proyectos de migración a código abierto**

---



---

HU 21	Gestionar proyectos de migración.	Alta	1	Debe permitir adicionar, editar, eliminar, mostrar detalles, listar y buscar los proyectos que se encuentran registrados en el sistema.
HU 22	Gestionar entidad cliente	Baja	2	Debe permitir adicionar, editar y mostrar detalles de las entidades registradas en el sistema.
HU 23	Gestionar contacto por parte del cliente.	Baja	2	Se gestiona el contacto por parte del cliente permitiendo listar, adicionar, editar y mostrar los detalles de los mismos. Además, permite asignar el contacto principal.
HU 24	Gestionar los departamentos de una entidad.	Baja	2	Debe permitir listar, adicionar, editar, eliminar y mostrar detalles de un departamento de una entidad seleccionada.
HU 26	Gestionar las tareas de un proyecto de migración.	Alta	1	Se gestionan las tareas de los proyectos, permitiendo listar, adicionar, editar, eliminar, buscar y mostrar detalles de las mismas.
HU 28	Gestionar fases del proyecto.	Baja	2	Se gestionan las fases del proyecto permitiendo adicionar, editar, eliminar y listar las fases del proyecto.
HU 29	Gestionar recurso de proyecto.	Alta	1	Se gestiona el recurso del proyecto, permitiendo adicionar, editar, eliminar, mostrar detalles, buscar y mostrar los recursos.
HU 33	Gestionar tipo de recurso	Alta	1	Se gestiona el tipo de recurso permitiendo buscar, adicionar, editar y eliminar los mismos.

## **2.2: Tecnologías y herramientas para el diseño e implementación**

### **Tecnologías**

Al igual que la metodología SXP, las herramientas y el resto de las tecnologías empleadas para el desarrollo de la solución propuesta son heredadas de la arquitectura que establece la PCMCA. Entre ellas se encuentran:

APACHE 2 como servidor web HTTP, de código abierto, ejecutable en varios sistemas operativos como: Unix, FreeBSD, Linux, Solaris, Novell NetWare, OS X, Microsoft Windows y que permite múltiples lenguajes de script como PHP, Perl, Tcl y Python (Mestras, 2013).

PostgreSQL 9.1 como gestor de bases de datos multiplataforma y con soporte nativo para varios lenguajes entre ellos PHP (Vitier y González, 2011).

PHP en su versión 5.3 como lenguaje de programación, interpretado de alto nivel embebido en páginas HTML y ejecutado en el servidor (Bakken, 2001).

Symfony 1.4.6 como *framework* desarrollado completamente con PHP 5, multiplataforma y compatible con la mayoría de los gestores de bases de datos incluyendo PostgreSQL (Potencier y Zaninotto, 2008).

Bootstrap 3 como *framework* para el desarrollo de aplicaciones web, sencillo y ligero basado en los últimos estándares de desarrollo como HTML5, CSS3 y JavaScript/Jquery. Compatible con los navegadores más usados (Mestras, 2014).

### **Herramientas**

Para el desarrollo y depuración del código se emplea el NetBeans 8.0 como entorno integrado de desarrollo (IDE) (Oracle, 2015), el pgAdmin 1.16 como administrador de bases de datos (pgAdmin, 2015), el Subversion 0.12 como sistema para el control de versiones (Ben Collins-Sussman, 2002), el TrueCrypt



7.1 para la codificación y seguridad del código (Gibson Research Corporation, 2014) y el Visual Paradigm for UML 8.0 para el diseño y modelado de las funcionalidades a implementar (Visual Paradigm International, 2015). Cada una de las herramientas mencionadas anteriormente fueron definidas para el desarrollo de la PCMCA, motivo por el cual es necesario emplearlas en el desarrollo de las funcionalidades a incorporar al sistema.

### **2.3: Modelo de diseño**

Como se menciona en el epígrafe 1.7, en el subsistema PCS predominan rasgos arquitectónicos de SOA y Presentación Desacoplada. En el Sistema de manejo de interfaces, como su nombre lo indica, es donde se encuentran las interfaces de cada uno de los subsistemas de la plataforma separados del flujo de datos y su manipulación. La comunicación entre estos componentes estará proporcionada por servicios, característica propia de SOA, Figura 5.

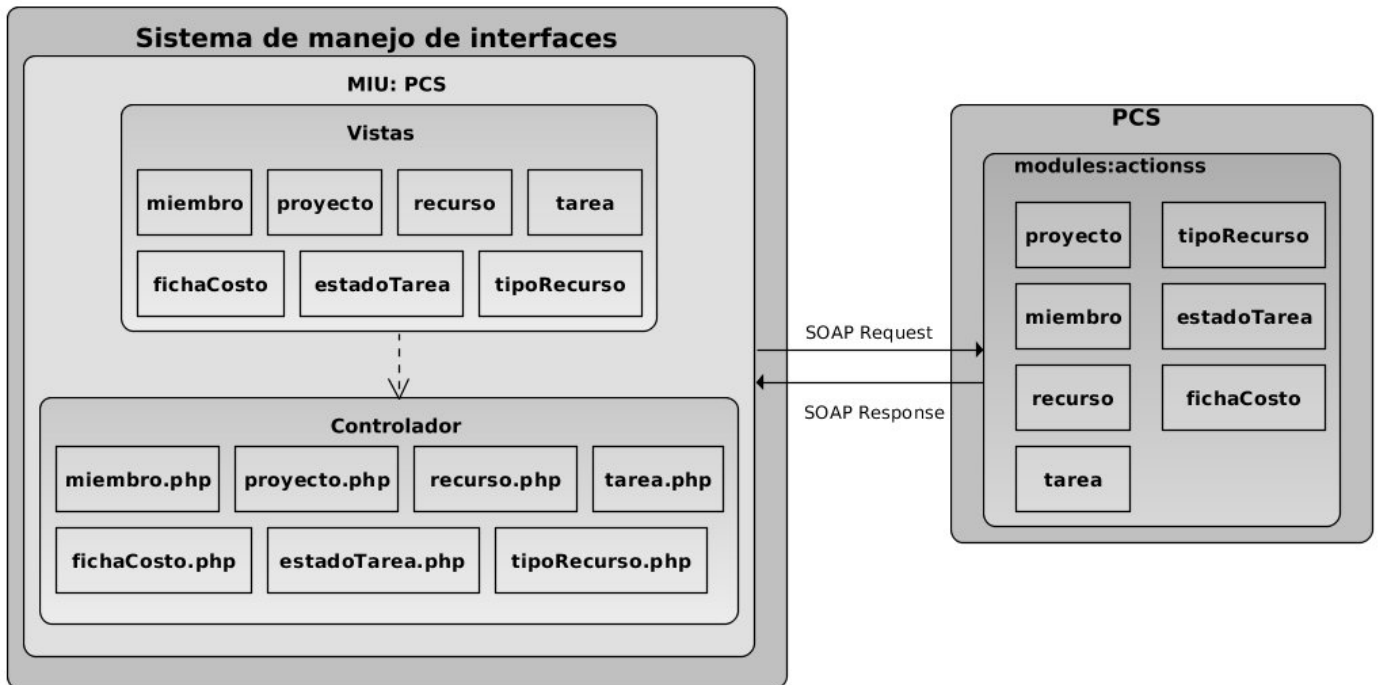


Figura 5: Arquitectura del subsistema PCS.

A pesar de predominar en la arquitectura del subsistema características de SOA y Presentación Desacoplada, el PCS internamente presenta rasgos estructurales afines al estilo arquitectónico N-capas (Vítier y González, 2011). Este permite la realización de actualizaciones en el interior de las capas, sin que esto afecte al resto del sistema y el hecho de distribuir las capas en distintos niveles físicos, mejora la escalabilidad y la tolerancia a fallos.

En la siguiente figura se muestra el diseño del subsistema en términos de clases y paquetes:



Como se muestra en la figura anterior, las clases se encuentran agrupadas en tres capas lógicas, siguiendo un estilo N-Capas que no entra en conflicto con los demás estilos utilizados en esta arquitectura. La capa presentación contiene la clase *ckSOAPHandler* que es la encargada de manejar el protocolo *Simple Object Access Protocol* (SOAP), traduciendo funcionalidades en servicios web. En la capa controlador se encuentran las clases que controlan la lógica del negocio y que se relacionan con las clases de la capa acceso a datos, lo que permite abstraerse del sistema gestor de bases de datos que se utilice.

### **Patrones de diseño**

Los patrones de diseño permiten la reutilización de las mejores prácticas y evitar errores graves, su utilidad se ha demostrado empíricamente. Mejoran además las propiedades de calidad de *software*, como mantenimiento y reutilización, acelerando el tiempo de desarrollo (Shehory y Sturm, 2014). Para la implementación de la solución propuesta se aplicaron aquellos definidos por el propio *framework* symfony, entre los que se encuentran:

#### **Patrones generales de *software* para asignar responsabilidades (GRASP)**

Los patrones GRASP (del inglés, *General Responsibility Assignment Software Patterns*) representan los principios básicos de la asignación de responsabilidades a objetos, expresados en forma de patrones. Dentro de dichos patrones son utilizados los que se mencionan a continuación:

Experto: Cada objeto es responsable por mantener su propia información y puede informar y modificar el valor de sus atributos (Cortés y Casallas, 2014). Un ejemplo de ello son las clases encargadas de hacer las consultas a la base de datos utilizando *Doctrine*, ya que tienen los atributos necesarios para efectuar dichas consultas y por tanto tienen la responsabilidad de realizar directamente las acciones con la base de datos.

Creador: todos los módulos del sistema tienen una clase *Actions* que contiene las acciones definidas para dichos módulos y es en ella donde se ejecutan los métodos que permiten el funcionamiento del sistema. En esta clase las acciones se encargan de crear los objetos de las clases que representan las entidades.

Alta Cohesión: garantiza que la información que almacena una clase debe ser coherente y estar relacionada con la clase a la que pertenece (Cortés y Casallas, 2014). Por ejemplo, la clase *proyectoActions* contiene varias funcionalidades en las que cada una cumple un objetivo específico y que solo el proyecto realiza.

Bajo acoplamiento: La clase *Actions* hereda únicamente de *sfActions* para alcanzar un bajo acoplamiento de clases. Las clases que implementan la lógica del negocio y de acceso a datos se encuentran en el modelo, las cuales no tienen asociaciones con las de la vista o el controlador, lo que proporciona que la dependencia en este caso sea baja.

A continuación se detallan las modificaciones realizadas en las clases de la capa controlador:

### 2.3.1: Clase *contactoActions*.

Tabla 3: Clase *contactoActions*

Método	Descripción	Parámetros	Valor de Retorno
listarContacto	Muestra una lista de contactos utilizando paginado, por lo que devuelve un conjunto de contactos a mostrar de la cantidad total.	\$pagina_identifica : int \$cantidad_por_pagina: int \$param_busq: string	contactoMeta (lista de contactos a mostrar y cantidad total de contactos).
insertarContacto	Añade un nuevo contacto al sistema.	\$nombre: string \$apellido: string	int (Identificador del contacto añadido).

**Capítulo 2: Diseño e implementación de las funcionalidades para la estimación de costos de proyectos de migración a código abierto**

---



---

		\$cargo: string \$direccion: string \$ciudad: string \$estado: string \$postal: string \$pais: string \$telefono1: string \$telefono2: string \$fax: string \$telefono_movil: string \$correo: string	
insertarContactosAlaEntidad	Inserta un nuevo contacto a la entidad.	\$id_contacto: int[] \$id_entidad: int[]	int (Identificador del contacto añadido en la entidad).
eliminarContactosDeLaEntidad	Elimina el contacto de la entidad.	\$id_entidad: int	bool ( <i>true</i> si se eliminaron).
editarContacto	Edita el contacto seleccionado.	\$id_contacto: int \$nombre: string \$apellido: string \$cargo: string \$direccion: string \$ciudad: string \$estado: string \$postal: string \$pais: string \$telefono1: string \$telefono2: string	bool ( <i>true</i> si se modificó).

		\$fax: string \$telefono_movil: string \$correo: string	
eliminarContacto	Elimina el contacto del sistema.	\$id_contacto: int[]	bool ( <i>true</i> si se elimina el contacto).
devolverContactoPorId	Obtiene un contacto del sistema a partir de su identificador.	\$id_contacto: int	contacto (el contacto obtenido).
devolverUltimoContacto	Obtiene el ultimo contacto del sistema.		contacto (el contacto obtenido).

### 2.3.2: Clase costosIndirectosActions.

Tabla 4: Clase costosIndirectosActions.

Método	Descripción	Parámetros	Valor de Retorno
editarRecursosIndirectos	Edita el recurso indirecto seleccionado.	\$id_proyecto: int \$id_recurso: int[] \$disponibilidad: int[] \$miembro: int[] \$horas_dedicadas: int[]	bool ( <i>true</i> si se modificó).
listarMiembrosIndirectos	Muestra una lista de miembros indirectos utilizando paginado, por lo que devuelve un conjunto de miembros indirectos a mostrar de la cantidad	\$pagina_identifica: int \$cantidad_por_pagina: int \$idproyecto: int	miembroIndirectoMeta (lista de miembros indirectos a mostrar y la cantidad total).

	total.		
listarRecursosPorRecursosIndirectos	Muestra una lista de recursos por recursos indirectos utilizando paginado, por lo que devuelve un conjunto de recursos indirectos a mostrar de la cantidad total.	\$pagina identifica: int \$cantidad_por_pagina: int \$idproyecto: int	recursoMeta (lista de recursos indirectos a mostrar y la cantidad total).

### 2.3.3: Clase departamentoActions

Tabla 5: Clase departamentoActions

Método	Descripción	Parámetros	Valor de Retorno
listardepartamento	Muestra una lista de departamentos utilizando paginado, por lo que devuelve un conjunto de departamentos a mostrar de la cantidad total.	\$pagina identifica: int \$cantidad_por_pagina: int	departamentoMeta (lista de departamentos a mostrar y la cantidad total).
insertardepartamento	Inserta un nuevo departamento al sistema.	\$nombre: string \$id_departamento_padre: int \$descripcion: string \$id_entidad: int \$jefe_departamento: int	int (identificador del departamento añadido al sistema).
eliminardepartamento	Elimina el departamento del	\$id_departamento: int	bool ( <i>true</i> si se



**Capítulo 2: Diseño e implementación de las funcionalidades para la estimación de costos de proyectos de migración a código abierto**

	sistema.		eliminaron).
editardepartamento	Edita el departamento seleccionado.	\$id_departamento: int \$nombre: string \$id_departamento_padre: int \$descripcion: string \$id_entidad: int \$jefe_departamento: int	bool ( <i>true</i> si se modificó).
devolverDepartamentoPorId	Obtiene un departamento del sistema a partir de su identificador.	\$id_departamento: int	departamento (el departamento obtenido).
existeDepartamentoEnEntidad	Busca si existe un departamento en la entidad.	\$id_entidad: int	bool ( <i>true</i> si se existe un departamento en la entidad).
devolverDepartamentoPorEntidad	Obtiene un departamento de la entidad a partir del identificador de la entidad.	\$id_entidad: int \$pagina_identifica: int \$cantidad_por_pagina: int	departamentoMeta (lista de departamentos a mostrar y la cantidad total).
dptolsHoja	Inserta el identificador del departamento que es hoja.	\$id Identificador: int	int (identificador del identificador añadido al sistema).
generateOrganigrama	Obtiene todas las entidades que pertenecen al departamento.	\$id_entidad: int	organoMeta (lista de entidades a mostrar y la cantidad total).

### 2.3.4: Clase dependenciaActions

Tabla 6: Clase dependenciaActions

Método	Descripción	Parámetros	Valor de Retorno
listarDependencia	Muestra una lista de dependencias utilizando paginado, por lo que devuelve un conjunto de dependencias a mostrar de la cantidad total.	\$pagina identifica: int \$cantidad_por_pagina: int	dependenciaMeta (lista de dependencias a mostrar y la cantidad total).

### 2.3.5: Clase faseActions.

Tabla 7: Clase faseActions.

Método	Descripción	Parámetros	Valor de Retorno
devolverNombreDeFasePor Prioridad	Obtiene una fase a partir de la prioridad.	\$prioridad: int	fase (la fase obtenida).

### 2.3.6: Clase fichaCostoActions.

Tabla 8: Clase fichaCostoActions.

Método	Descripción	Parámetros	Valor de Retorno
listarFichaCosto	Muestra una lista de coeficientes utilizando paginado, por lo que devuelve un conjunto de	\$param_busq: int \$pagina identifica: int \$cantidad_por_pagina: int	fichaCostoMeta (lista de coeficientes a mostrar y la cantidad total).

**Capítulo 2: Diseño e implementación de las funcionalidades para la estimación de costos de proyectos de migración a código abierto**

	coeficientes a mostrar de la cantidad total.		
editarFichaCosto	Edita el coeficiente seleccionado.	\$id_elemento: string \$monto: float	bool ( <i>true</i> si se modificó).
devolverMontoPorId	Obtiene un monto de la ficha de costos a partir de su identificador.	\$id_elemento: string	fichaCosto (el monto obtenido).
insertarTendenciaCosto	Añade una nueva tendencia al sistema.	\$id_proyecto: int \$fecha_variable: string \$presupuesto_real: float \$costo_proyecto: float	int (identificador de la tendencia añadida).
listarTendenciaCostoPorProyecto	Muestra una lista de tendencias pertenecientes a un proyecto utilizando paginado, por lo que devuelve un conjunto de tendencias a mostrar de la cantidad total.	\$pagina_identifica: int \$cantidad_por_pagina: int \$id_proyecto: int	tendenciaCostoMeta (lista de tendencias a mostrar y cantidad total de tendencias).
editarTendenciaCosto	Edita la tendencia seleccionada.	\$id_proyecto: int \$fecha_variable: string \$presupuesto_real: float \$costo_proyecto: float	bool ( <i>true</i> si se modificó).
ultimaTendenciaCosto	Obtiene la última tendencia del sistema.		tendenciaCosto (la tendencia obtenida).

### 2.3.7: Clase miembroActions

Tabla 9: Clase miembroActions

Método	Descripción	Parámetros	Valor de Retorno
insertarMiembro	Añade un nuevo miembro al sistema.	\$nombre nombre: string \$apellido: string \$telefono1: string \$telefono_movil: string \$correo: string \$sid_valor_peso_hora: float	int (identificador del miembro añadido).
editarMiembro	Edita el miembro seleccionado.	\$id_miembro: int \$nombre nombre: string \$apellido: string \$telefono1: string \$telefono_movil: string \$correo: string \$sid_valor_peso_hora: float	bool ( <i>true</i> si se modificó).

### 2.3.8: Clase proyectoActions.

Tabla 10: Clase proyectoActions.

Método	Descripción	Parámetros	Valor de Retorno
listarRecursProyecto	Muestra una lista de recursos utilizando paginado, por lo que devuelve un conjunto de recursos a mostrar de la	\$id_proyecto: int	recursoMeta (lista de recursos a mostrar y cantidad total de recursos).

**Capítulo 2: Diseño e implementación de las funcionalidades para la estimación de costos de proyectos de migración a código abierto**

	cantidad total.		
insertarProyecto	Añade un nuevo proyecto al sistema.	\$nombre: string \$descripcion: string \$nombre_oficial: string \$fecha_inicio: string \$fecha_fin: string \$presupuesto_tentativo: float \$presupuesto_real: float \$alcance_proyecto: string \$objetivo_general: string \$objetivo_especifico: string \$impacto_proyecto: string \$es_proyecto_tipo: int \$id_prioridad: int \$id_estadoProyecto: int \$id_modalidad: int \$id_recurso: int[] \$disponibilidad_recurso: int[] \$miembro: int[] \$rol: int[] \$idProyectoReferencia: int \$importar: string[]	int (identificador del proyecto añadido).
insertarProyectoAPartirDe ProyectoTipo	Añade un nuevo proyecto al sistema a partir de un proyecto tipo.	\$nombre: string \$descripcion: string \$nombre_oficial: string \$fecha_inicio: string \$fecha_fin: string	int (identificador del proyecto añadido).

**Capítulo 2: Diseño e implementación de las funcionalidades para la estimación de costos de proyectos de migración a código abierto**

---



---

		<p>\$presupuesto_tentativo: float          \$presupuesto_real: float          \$alcance_proyecto: string          \$objetivo_general: string          \$objetivo_especifico: string          \$impacto_proyecto: string          \$responsable_desarrollador: string          \$id_prioridad: int          \$id_estadoProyecto: int          \$id_modalidad: int          \$responsable_cliente: int</p>	
eliminarProyecto	Elimina el proyecto del sistema.	\$id_proyecto: int [ ]	bool ( <i>true</i> si se eliminaron).
editarProyecto	Edita el proyecto seleccionado.	<p>\$id_proyecto: int          \$nombre: string          \$descripcion: string          \$nombre_oficial: string          \$fecha_inicio: string          \$fecha_fin: string          \$presupuesto_tentativo: float          \$presupuesto_real: float          \$alcance_proyecto: string          \$objetivo_general: string          \$objetivo_especifico: string          \$impacto_proyecto: string          \$es_proyecto_tipo: int          \$id_prioridad: int</p>	bool ( <i>true</i> si se modificó).

**Capítulo 2: Diseño e implementación de las funcionalidades para la estimación de costos de proyectos de migración a código abierto**

		\$id_estadoProyecto: int \$id_modalidad: int \$id_recurso: int[] \$disponibilidad_recurso: int[] \$miembro: int[] \$rol: int[] \$idProyectoReferencia: int \$importar: string[]	
insertarMiembroAlProyecto	Añade un nuevo miembro al proyecto.	\$id_miembro: int \$id_proyecto: int	int (identificador del miembro añadido).
eliminarMiembrosDelProyecto	Elimina el miembro del proyecto.	\$id_proyecto: int	bool ( <i>true</i> si se eliminaron).
listarMiembroPorProyecto	Muestra una lista de miembros por proyecto utilizando paginado, por lo que devuelve un conjunto de miembros a mostrar de la cantidad total.	\$pagina: int \$cantidad_por_pagina: int \$id_proyecto: int	miembroMeta (lista de miembros a mostrar y cantidad total de miembros).
listarRecursosPorProyecto	Muestra una lista de recursos por proyecto utilizando paginado, por lo que devuelve un conjunto de recursos a mostrar de la cantidad total.	\$pagina: int \$cantidad_por_pagina: int \$id_proyecto: int	recursoMeta (lista de recursos a mostrar y cantidad total de recursos).
insertarRecursoAlProyecto	Añade un nuevo recurso	\$id_recurso: int \$id_proyecto: int	int (identificador del

**Capítulo 2: Diseño e implementación de las funcionalidades para la estimación de costos de proyectos de migración a código abierto**

	al proyecto.		recurso añadido).
listarContactoPorProyecto	Muestra una lista de contactos por proyecto utilizando paginado, por lo que devuelve un conjunto de contactos a mostrar de la cantidad total.	\$pagina: int \$cantidad_por_pagina: int \$id_proyecto: int	contactoMeta (lista de contactos a mostrar y cantidad total de contactos).
actualizarContactoCliente	Actualiza el contacto de la entidad cliente del proyecto.	\$id_proyecto: int \$idcontacto: int	bool ( <i>true</i> si se modificó).
editarRecursosAgotados	Edita el recurso agotado seleccionado.	\$id_proyecto: int	bool ( <i>true</i> si se modificó).
actualizarResponsableProyecto	Actualiza el responsable del proyecto.	\$id_proyecto: int \$id_miembro: int	bool ( <i>true</i> si se modificó).

### 2.3.9: Class recursoActions

Tabla 11: Class recursoActions

Método	Descripción	Parámetros	Valor de Retorno
listarRecurso	Muestra una lista de recursos utilizando paginado, por lo que devuelve un conjunto de recursos a mostrar de la cantidad total.	\$pagina: int \$cantidad_por_pagina: int \$pagina identifica: int	recursoMeta (lista de recursos a mostrar y cantidad total de recursos).



insertarRecurso	Añade un nuevo recurso.	\$nombre: string \$notas: string \$costo: float \$tipo_recurso: int \$disponibilidad: int \$tasa_depreciacion: float \$inventario: string \$cant_x_paquete: int	int (identificador del recurso añadido).
editarRecurso	Edita el recurso del sistema.	\$id_recurso: int \$nombre: string \$notas: string \$costo: float \$tipo_recurso: int \$disponibilidad: int \$tasa_depreciacion: float \$inventario: string \$cant_x_paquete: int	bool ( <i>true</i> si se modificó).
devolverRecursoPorId	Devuelve un recurso por su identificador.	\$id_recurso: int	int (identificador del recurso añadido).

### 2.3.10: Class tareaActions

Tabla 12: Class tareaActions

Método	Descripción	Parámetros	Valor de Retorno
listarTarea	Muestra una lista de tareas utilizando	\$pagina: int \$cantidad_por_pagina: int	tareaMeta (lista de tareas a mostrar y cantidad total)

**Capítulo 2: Diseño e implementación de las funcionalidades para la estimación de costos de proyectos de migración a código abierto**

	paginado, por lo que devuelve un conjunto de tareas a mostrar de la cantidad total.		de tareas).
listarRecursTarea	Muestra una lista de recursos al adicionar las tareas utilizando paginado, por lo que devuelve un conjunto de recursos a mostrar de la cantidad total.	\$id_tarea: int	recursoMeta (lista de recursos a mostrar y cantidad total de recursos).
listarRecursosPorTarea	Muestra una lista de recursos de las tareas utilizando paginado, por lo que devuelve un conjunto de recursos a mostrar de la cantidad total.	\$pagina: int \$cantidad_por_pagina: int \$id_tarea: int \$id_proyecto: int	recursoMeta (lista de recursos a mostrar y cantidad total de recursos).
listarMiembrosPorProyecto	Muestra una lista de miembros por proyecto utilizando paginado, por lo que devuelve un conjunto de miembros a mostrar de la cantidad total.	\$id_proyecto: int \$pagina: int \$cantidad_por_pagina: int \$param_busq: string	miembroMeta (lista de miembros a mostrar y cantidad total de miembros).
listarMiembroPorTarea	Muestra una lista de	\$pagina: int	tareaMiembroProyectoM

**Capítulo 2: Diseño e implementación de las funcionalidades para la estimación de costos de proyectos de migración a código abierto**

	miembros de las tareas utilizando paginado, por lo que devuelve un conjunto de miembros a mostrar de la cantidad total.	\$cantidad_por_pagina: int \$id_tarea: int \$id_proyecto: int	eta (lista de miembros que pertenecen al proyecto a mostrar y cantidad total de recursos).
devolverTareasPorPrioridad	Devuelve tareas por el identificador de la prioridad.	\$id_prioridad: int \$pagina: int \$cantidad_por_pagina: int \$cantidad_por_pagina: int	tareaMeta (lista de tareas a mostrar y cantidad total de tareas).
devolverTareaPorId	Devuelve tareas por el identificador.	\$id_tarea: int	tarea(la tarea obtenido).
devolverTareasPorProyecto	Devuelve tareas por el identificador del proyecto.	\$id_proyecto: int \$pagina: int \$cantidad_por_pagina: int \$cantidad_por_pagina: int	tareaMeta (lista de tareas a mostrar y cantidad total de tareas).
existeTareaEnProyecto	Busca si existe determinada tarea en el proyecto.	\$id_proyecto: int	bool ( <i>true</i> si se existe una tarea en el proyecto).
existeTarea	Busca si existe una tarea.	\$id_proyecto: int \$id_tarea: int	bool ( <i>true</i> si se existe una tarea).
editarAgotadosResources	Edita el recurso agotado de la tarea.	\$id_proyecto: int \$id_tarea: int	bool ( <i>true</i> si se modificó).
insertarTarea	Añade una nueva tarea.	\$id_proyecto: int \$nombre: string	int (identificador de la tarea añadida).

**Capítulo 2: Diseño e implementación de las funcionalidades para la estimación de costos de proyectos de migración a código abierto**

---



---

		\$id_tipo_tarea: int \$id_estadoTarea: int \$fecha_inicio: string \$fecha_fin: string \$descripcion: string \$id_flujo: int \$id_fase: int \$id_prioridad: int \$progreso: int \$presupuesto: float \$evaluacion: int \$id_tarea_padre: int \$id_relacion: string \$id_dependencia: int \$miembro: int [ ] \$horas_dedicadas: int [ ] \$id_recurso: int [ ] \$disponibilidad: int [ ]	
editarTarea	Edita la tarea del proyecto.	\$id_tarea: int \$id_proyecto: int \$nombre: string \$id_tipo_tarea: int \$id_estadoTarea: int \$fecha_inicio: string \$fecha_fin: string \$descripcion: string \$id_flujo: int	bool ( <i>true</i> si se modificó).

		\$id_fase: int \$id_prioridad: int \$progreso: int \$presupuesto: float \$evaluacion: int \$id_tarea_padre: int \$id_relacion: string \$id_dependencia: int \$miembro: int [ ] \$horas_dedicadas: int [ ] \$id_recurso: int [ ] \$disponibilidad: int [ ]	
eliminarTarea	Elimina la tarea seleccionada del proyecto.	\$id_tarea: int [ ] \$id_proyecto: int	bool ( <i>true</i> si se elimina).
gantFlujo	Devuelve tareas por el identificador del proyecto.	\$id_proyecto: int \$pagina: int \$param_busq: string	tareaMeta (lista de tareas a mostrar y cantidad total de tareas).

### 2.3.11: Class tipoRecursoActions

Tabla 13: Class tipoRecursoActions

Método	Descripción	Parámetros	Valor de Retorno
listarTipoRecurso	Muestra una lista de tipos de recursos utilizando paginado, por	\$param_busq: string \$pagina identifica: int \$cantidad_por_pagina: int	TipoRecursoMeta (lista de tareas a mostrar y cantidad total de tareas).

	lo que devuelve un conjunto de tipos de recursos a mostrar de la cantidad total.		
insertarTipoRecurso	Añade un nuevo tipo de recurso.	\$descripcion: string \$porcentaje: float	int (identificador del tipo de recurso añadido).
eliminarTipoRecurso	Elimina el tipo de recurso seleccionado.	\$id_tipo_recurso: int	bool ( <i>true</i> si se eliminaron).
editarTipoRecurso	Edita el tipo de recurso seleccionado.	\$id_tipo_recurso: int \$descripcion: string \$porcentaje: float	bool ( <i>true</i> si se modificó).
devolverTipoRecursoPorId	Devuelve el tipo de recurso por identificador.	\$id_tipo_recurso: int	tipoRecursoMeta (lista de tareas a mostrar y cantidad total de tareas).

## 2.4: Estructura de componentes

En la siguiente figura se muestran los principales componentes que intervienen en la implementación. Asociados a los servicios que permiten la comunicación entre las interfaces y la lógica de negocio están los componentes *pcsWS.php* y los recogidos en *ckWebServicesPlugin*. Por otro lado se encuentran las librerías donde están incluidas *modelWS* con la implementación de cada uno de los objetos que intervienen en la lógica de negocio y *Servicios* que contiene el código de cada uno de los métodos invocados en las clases *Actions*. Estas últimas están incluidas en los módulos del sistema recogidos en el paquete *modules*. Los archivos *.yml* ubicados en el paquete *config* permiten personalizar cualquier aspecto de *symfony* y añadir parámetros específicos para la propia aplicación. En el caso de *schema.yml*,

**Capítulo 2: Diseño e implementación de las funcionalidades para la estimación de costos de proyectos de migración a código abierto**

---

---

es el archivo que contiene la representación del modelo de datos relacional del proyecto y el que permite el acceso a los datos.

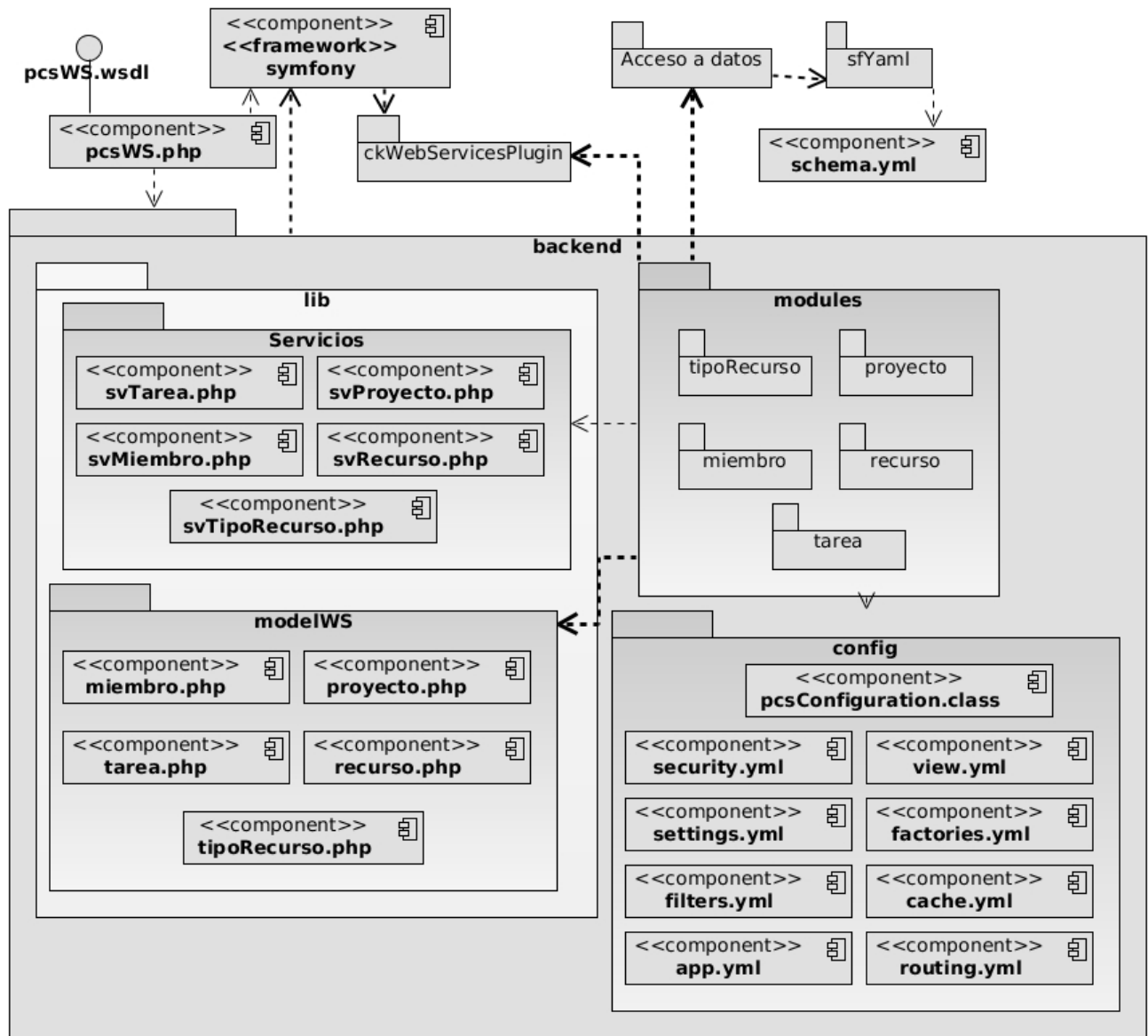


Figura 7: Componentes del sistema.



## Capítulo 3: Validación de la solución propuesta

Concluida la etapa de implementación es necesario verificar que el sistema cumple con los requisitos que exige el cliente y posee un correcto funcionamiento. Para ello se describen: los resultados obtenidos una vez aplicadas las pruebas de aceptación; la eficiencia lograda al generar una ficha de costos mediante la operacionalización de las variables; y el nivel de satisfacción de los especialistas de migración haciendo uso de la técnica de V. A. Iadov (Cuscó y Wells, 2008).

### 3.1: Casos de Prueba de Aceptación

Caso de prueba de aceptación es el artefacto escogido por la metodología SXP para documentar las pruebas de caja negra a realizar a las funcionalidades implementadas. Permite validar que un sistema cumple con el funcionamiento esperado y permite que el usuario de dicho sistema determine su aceptación, desde el punto de vista de su funcionalidad y rendimiento (Céspedes, García y Romero, 2008).

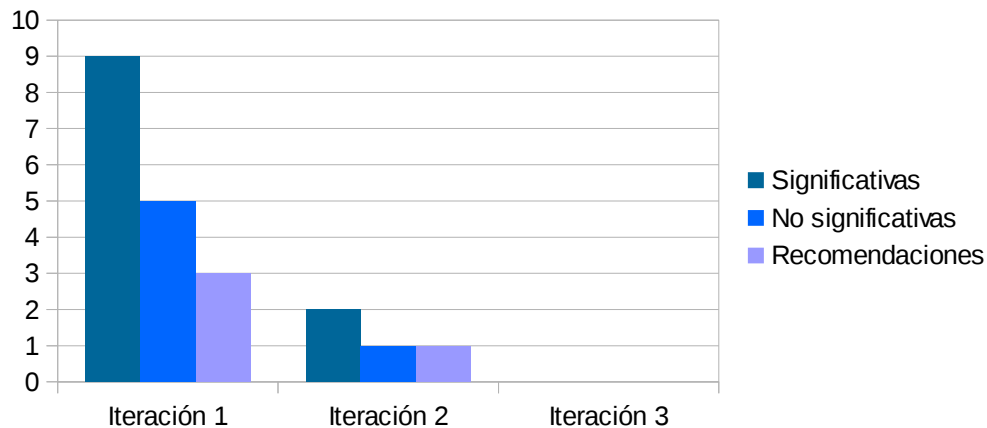
A continuación se muestra la estructura de un Caso de prueba de aceptación usando como ejemplo el caso correspondiente a la HU: *Generar ficha de costos de un proyecto*.

Tabla 14: Caso de prueba de aceptación: Generar ficha de costos de un proyecto

Caso de Prueba de Aceptación	
<b>Código Caso de Prueba:</b> PCMCA-06-01	<b>Nombre Historia de Usuario:</b> Generar ficha de costos de un proyecto.
<b>Nombre de la persona que realiza la prueba:</b> Jailen García González	
<b>Descripción de la Prueba:</b> Prueba a la funcionalidad generar <b>ficha de costos</b> .	
<b>Condiciones de Ejecución:</b> Haberse autenticado con una identidad que tenga permisos para generar la ficha de costos de un proyecto. Acceder a la interfaz del módulo de PCS, a la funcionalidad Proyectos y acceder a proyecto de migración en detalles. Debe haber insertado en el sistema al menos un proyecto.	
<b>Entrada / Pasos de ejecución:</b>	

<b>1. Dar clic sobre ficha de costos</b>
<b>Resultado Esperado:</b> Se muestra la ficha de costos en forma de tabla.
<b>Evaluación de la Prueba:</b> Satisfactoria

Las pruebas se realizaron en 3 iteraciones con la participación del cliente, donde fueron arrojados los siguientes resultados:



Gráfica 1: No conformidades encontradas en cada iteración de pruebas.

De esta manera queda reflejada la aceptación del cliente en cuanto a las funcionalidades desarrolladas y el correcto funcionamiento de las mismas.

### 3.2: Técnica de V. A. Iadov

Tratándose de un sistema que será usado para la gestión de proyectos de migración, se realiza un estudio para determinar el nivel de satisfacción de los especialistas de migración con los resultados obtenidos. La técnica de Iadov constituye una vía indirecta para el estudio de la satisfacción, ya que los criterios que se utilizan se fundamentan en las relaciones que se establecen entre tres preguntas cerradas

que se intercalan dentro de un cuestionario y cuya relación el sujeto desconoce (Cuscó y Wells, 2008). Estas tres preguntas se relacionan a través de lo que se denomina el “Cuadro Lógico de Iadov”, Tabla 15. La encuesta realizada puede consultarse en el Anexo 3.

Tabla 15: Cuadro Lógico de Iadov

¿Satisface sus necesidades como especialista las funcionalidades incluidas en el subsistema PCS para el proceso de estimación de costos de un proyecto?	¿Considera usted que se deba realizar el proceso de estimación de costos de un proyecto de migración a código abierto de forma manual, cuando se cuenta con la Plataforma Cubana de Migración a Código Abierto que recoge todos los datos necesarios para este proceso?								
	No			No sé			Sí		
	¿Si usted fuera a realizar el proceso de estimación de costos, lo realizaría utilizando las nuevas funcionalidades incluidas en el subsistema PCS?								
	Sí	No sé	No	Sí	No sé	No	Sí	No sé	No
Me gusta mucho	1	2	6	2	2	6	6	6	6
Me gusta más de lo que me disgusta	2	2	3	2	3	3	6	3	6
Me da lo mismo	3	3	3	3	3	3	3	3	3
Me disgusta más de lo que me gusta	6	3	6	3	4	4	3	4	4
No me gusta nada	6	6	6	6	4	4	6	4	5
No sé que decir	2	3	6	3	3	3	6	3	4

El número resultante de la interrelación de las tres preguntas indica la posición de cada sujeto en la escala de satisfacción (Cuscó y Wells, 2008).

La escala de satisfacción es la siguiente:

1. Máxima satisfacción.

2. Más satisfecho que insatisfecho.
3. No definida.
4. Más insatisfecho que satisfecho.
5. Máxima insatisfacción.
6. Contradictoria.

Para la realización de este estudio se contó con la participación de 11 especialistas de migración pertenecientes al centro CESOL de la UCI. Realizadas las encuestas, la interrelación de las preguntas respondidas por los encuestados se muestra en la siguiente tabla:

Tabla 16: Resultados de la encuesta

<b>Nivel de satisfacción</b>	<b>Cantidad de especialistas</b>
Máxima satisfacción.	8
Más satisfecho que insatisfecho.	3
No definida.	0
Más insatisfecho que satisfecho.	0
Máxima insatisfacción.	0
Contradictoria.	0

A continuación se procede a calcular el Índice de satisfacción grupal (ISG) convirtiendo la escala de satisfacción a una escala numérica, Tabla 17.

Tabla 17: Niveles de satisfacción en escala numérica

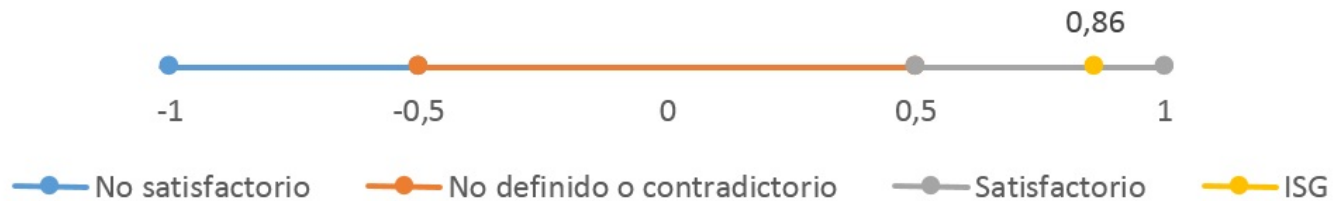
+1	Máximo de satisfacción
+0,5	Más satisfecho que insatisfecho
0	No definido o contradictorio
-0,5	Más insatisfecho que satisfecho
-1	Máxima insatisfacción

El ISG se calcula mediante la siguiente fórmula, donde A, B, C, D y E representan la cantidad de especialistas que corresponden a la clasificación asociada al coeficiente que lo multiplica.

$$ISG = A(+1) + B(+0,5) + C(0) + D(-0,5) + E(-1)$$

Valores sustituidos:  $ISG = 8*(+1) + 3*(+0,5) + 0*(0) + 0*(-0,5) + 0*(-1)$

Siendo  $ISG=0,86$  se considera que los especialistas de migración quedaron satisfechos con las nuevas funcionalidades implementadas en el subsistema PCS, ya que su valor está comprendido en el intervalo entre 0,5 y 1 como muestra la siguiente gráfica:



Gráfica 2: Intervalos de los niveles de satisfacción.

### 3.3: Operacionalización de las variables

Luego de verificar el correcto funcionamiento de la solución implementada, es necesario comprobar la eficiencia lograda a través de una comparación al realizar el proceso de estimación de costos de forma manual y posteriormente usando las nuevas funcionalidades, usando los datos de un mismo proyecto de migración. Para ello se aplica la operacionalización de las variables, la cual consiste en sustituir unas variables por otras más concretas, describiendo las operaciones que hay que realizar para medirlas, convirtiéndolas en indicadores observables y cuantificables. Con la operacionalización de las variables, se pasa de variables conceptuales a operativas y de estas a indicadores observables (León y González, 2011).

En la siguiente tabla se muestran los valores adquiridos por los indicadores definidos para la variable independiente:

Tabla 18: Medición de los indicadores definidos para la variable independiente.

Indicadores	Valor
Grado de satisfacción	Satisfecho
Resultado de las pruebas de aceptación	Satisfactoria

Para probar que las funcionalidades implementadas reducen el tiempo y los recursos necesarios para la realización del proceso de estimación, se miden ambos aspectos para un mismo proyecto de migración, en diferentes condiciones: primeramente de forma manual como se ha desarrollado hasta el momento y luego utilizando el subsistema PCS. El proyecto de migración escogido para el estudio es el correspondiente a la empresa *BK Import/Export* con una duración de 3 meses, para el cual se asignaron 2 recursos humanos y una computadora de escritorio para el desarrollo de las tareas, asumiendo además las personas asociadas a la administración del centro que inciden en dicho proyecto.

Luego de haber realizado el proceso de estimación de costos en ambos momentos, se obtuvieron los siguientes datos:

Tabla 19: Medición de los indicadores definidos para la variable dependiente.

<b>Indicadores</b>	<b>Manual</b>	<b>Desde la PCMCA</b>
Duración del proceso de estimación de costos.	30 minutos	4 segundos
Cantidad de recursos humanos necesarios para el proceso de estimación de costos.	4	1
Cantidad de recursos materiales necesarios para el proceso de estimación de costos.	1 computadora	1 computadora

Partiendo de los datos reflejados se hace evidente la eficiencia alcanzada al realizar la estimación de costos desde la PCMCA, siendo reducidos la cantidad de recursos humanos y el tiempo de duración del proceso.

Una vez realizadas las pruebas de aceptación, aplicada la técnica de V. A. Iadov y la operacionalización de las variables se concluye que el subsistema PCS está en condiciones para realizar el proceso de estimación de costos y generar la ficha de costos de un proyecto de manera eficiente.

## **Conclusiones**

1. Las funcionalidades para la generación de fichas de costo implementadas en el subsistema PCS de la PCMCA contribuyen a estimar los costos de los proyectos de migración a código abierto en menos tiempo y con menos recursos.
2. Los sistemas de gestión de costos Openbravo, Odoo y GESPRO, no satisfacen los requisitos del cliente; por lo que la modificación del subsistema PCS constituye la solución más adecuada.
3. Las funcionalidades diseñadas cumplen con las restricciones impuestas por la arquitectura de la PCMCA, integrándose correctamente.
4. El ISG calculado y las pruebas de aceptación realizadas, demuestran que las modificaciones realizadas al subsistema PCS satisfacen los requisitos del cliente.



## **Recomendaciones**

1. Incluir funcionalidades que permitan realizar un análisis en función de optimizar los costos de un proyecto de migración a código abierto.
2. Implementar funcionalidades que permitan calcular los coeficientes derivados del presupuesto dedicado a la producción.

## Referencias bibliográficas

1. BAKKEN, S.S., 2001. *Manual de PHP*. 2001. S.l.: s.n.
2. BANERJEE, B. 2009. *Cost accounting: Theory and Practice*. 12. New Delhi: PHI Learning.
3. BEN COLLINS-SUSSMAN 2002. Subversion Dev: Subversion 0.12.0 released. [en línea]. [Consulta: 10 junio 2015]. Disponible en: <http://svn.haxx.se/dev/archive-2002-05/0168.shtml>.
4. CENTRO DE EXCELENCIA EN COMERCIO ELECTRÓNICO 2011. 10 Programas ERP Software Libre y gratis para PYMES. *Comercio Electrónico Global* [en línea]. [Consulta: 1 junio 2015]. Disponible en: <http://www.e-global.es/erp/10-programas-erp-software-libre-y-gratis-para-pymes.html>.
5. CÉSPEDES, R.F., GARCÍA, S.P. y ROMERO, G.M.P., 2008. *Propuesta de un expediente, para los proyectos productivos del Polo de Software Libre, de la Facultad 10. SXP*. 2008. S.l.: s.n.
6. CORTÉS, G. y CASALLAS, R. 2014. Introducción a los patrones de Software. [en línea]. Universidad de los Andes. [Consulta: 12 mayo 2015]. Disponible en: <https://sistemas.uniandes.edu.co/~isis2701/dokuwiki/lib/exe/fetch.php?media=isis2701-patronesgrasp.pdf>.
7. CURTO, J.R.P. 2013. *BPM (Business Process Management): Cómo alcanzar la agilidad y eficiencia operacional a través de BPM y la empresa orientada a procesos* [en línea]. S.l.: BPMteca.com. [Consulta: 6 junio 2015]. ISBN 8461638549, 9788461638543. Disponible en: <https://books.google.com/cu/books?id=07NJBAAAQBAJ&pg=PA286&dq=m%C3%A9todos+estimaci%C3%B3n+costos&hl=es-419&sa=X&ei=nPVxVbuvDraTsQT6vr2oDA&ved=0CC0Q6AEwAw#v=onepage&q=m%C3%A9todos%20estimaci%C3%B3n%20costos&f=false>.
8. CUSCÓ, M.B.I. y WELLS, J.G., 2008. *La satisfacción del profesor de Educación Física*. 2008. S.l.: Funámbulos Editores.
9. DIOSBEL PÉREZ, YOENCY LEYVA, LIANNE REYES, ALEXANDER MARTÍNEZ y YASIEL PÉREZ 2014. *Plataforma Cubana de Migración a Código Abierto*. , vol. 8. ISSN 2227 - 1899.
10. GIBSON RESEARCH CORPORATION 2014. GRC's | TrueCrypt, the final release, archive. *Gibson Research Corporation* [en línea]. [Consulta: 11 junio 2015]. Disponible en: <https://www.grc.com/misc/truecrypt/truecrypt.htm>.
11. JIMÉNEZ, M.P.E.B. 2014. *Diccionario de Administración y Finanzas* [en línea]. S.l.: Palibrio. [Consulta: 5 junio 2015]. ISBN 1463354940, 9781463354947. Disponible en: <https://books.google.com/cu/books?>

id=8q0KBAAQBAJ&dq=definici%C3%B3n+costo&hl=es&source=gbs\_navlinks\_s.

12. JUAN, M.I., COSOLLAR, A.D., DOLZ, C.D. y ORTEGA, C.F. 2014. *Fundamentos de dirección de empresas. Conceptos y habilidades directivas* [en línea]. 2. Madrid, España: Ediciones Paraninfo, S.A. [Consulta: 11 junio 2015]. ISBN 8428399220, 9788428399227. Disponible en: <https://books.google.com.cu/books?id=X9v7CAAQBAJ&pg=PA20&dq=eficiencia+concepto&hl=es-419&sa=X&ved=0CCsQ6AEwA2oVChMI-K28ybuGxgIVIS2MCh2wBADE#v=onepage&q=eficiencia%20concepto&f=false>.
13. LEÓN, R.A.H. y GONZÁLEZ, S.C. 2011. *El Proceso de Investigación Científica*. Ciudad de La Habana, Cuba: Editorial Universitaria. ISBN 978-959-16- 1307-3.
14. LÓPEZ, F.J.T. 2013. *Administración de proyectos de informática* [en línea]. S.l.: ECOE EDICIONES. [Consulta: 5 junio 2015]. ISBN 9586488160, 9789586488167. Disponible en: [https://books.google.com.cu/books?id=vQFaAQAAQBAJ&dq=costos+indirectos&hl=es&source=gbs\\_navlinks\\_s](https://books.google.com.cu/books?id=vQFaAQAAQBAJ&dq=costos+indirectos&hl=es&source=gbs_navlinks_s).
15. MESTRAS, J.P. 2013. *ServidoresWeb-Apache*. [en línea]. [Consulta: 23 abril 2015]. Disponible en: <http://www.fdi.ucm.es/profesor/jpavon/web/31-ServidoresWeb-Apache.pdf>.
16. MESTRAS, J.P., 2014. *Bootstrap 3.0 Aplicaciones Web/ Sistemas Web*. 2014. S.l.: s.n.
17. OPENBRAVO, 2014. *La plataforma para comercio de Openbravo. Solución flexible de comercio para minoristas ágiles* [en línea]. 2014. S.l.: s.n. [Consulta: 25 enero 2015]. Disponible en: <http://www.openbravo.com/content/pdf/openbravo-commerce-platform-brochure-es-high.pdf>.
18. ORACLE 2015. *NetBeans IDE*. *NetBeans IDE* [en línea]. [Consulta: 10 junio 2015]. Disponible en: <https://netbeans.org/features/index.html>.
19. ORTIZ, S.S., BENITEZ, A.P. y HERNÁNDEZ, D.L.G., 2015. *Estrategia de soporte técnico para el proceso de migración a software libre en Cuba*. Abril 2015. S.l.: s.n.
20. PÉREZ, D.P.Y.P., 2013. *GESPRO. Paquete para la gestión de proyectos*. 2013. S.l.: s.n.
21. PGADMIN 2015. *pgAdmin 1.16 Visual Tour*. *pgAdmin PostgreSQL Tools* [en línea]. [Consulta: 10 junio 2015]. Disponible en: <http://www.pgadmin.org/visualtour16.php>.
22. PMI 2013. *Guía de los fundamentos para la dirección de proyectos* [en línea]. Quinta. Estados Unidos: s.n. ISBN 978-1-62825-009-1. Disponible en: [http://portales.puj.edu.co/wjfajardo/ADMINISTRACION%20DE%20OBRAS/PRESUPUESTO/3-%20ANALISIS%20ESTRATEGICO/03%20Material%20de%20Referencia/PMBOK\\_Guide5th\\_Spanish.pdf](http://portales.puj.edu.co/wjfajardo/ADMINISTRACION%20DE%20OBRAS/PRESUPUESTO/3-%20ANALISIS%20ESTRATEGICO/03%20Material%20de%20Referencia/PMBOK_Guide5th_Spanish.pdf).

23. POTENCIER, F. y ZANINOTTO, F., 2008. *Symfony la guía definitiva*. 2008. S.l.: s.n.
24. RICH, J., MOWEN, M., HANSEN, D. y JONES, J. 2013. *Cornerstones of Financial Accounting* [en línea]. 3. S.l.: Cengage Learning. [Consulta: 5 junio 2015]. ISBN 1285607155, 9781285607153. Disponible en: [https://books.google.com/cu/books?id=jbiWAAAAQBAJ&dq=depreciation+methods&hl=es&source=gbs\\_navlinks\\_s](https://books.google.com/cu/books?id=jbiWAAAAQBAJ&dq=depreciation+methods&hl=es&source=gbs_navlinks_s).
25. ROMERO, G.M.P., 2008. *SXP. Metodología ágil para proyectos de software libre*. octubre 2008. S.l.: s.n.
26. SÁNCHEZ, R.R. 2015. Modelos de Negocio en Software Libre y los ERP: OpenBravo » Software Libre y Comunicación. *Software Libre y Comunicación* [en línea]. [Consulta: 1 junio 2015]. Disponible en: <http://ramonramon.org/blog/2015/01/14/modelos-de-negocio-en-software-libre-y-los-erp-openbravo/>.
27. SHEHORY, O. y STURM, A. (eds.) 2014. *Agent-Oriented Software Engineering: Reflections on Architectures, Methodologies, Languages, and Frameworks* [en línea]. S.l.: Springer. [Consulta: 5 junio 2015]. ISBN 3642544320, 9783642544323. Disponible en: [https://books.google.com/cu/books?id=2pjlAwAAQBAJ&dq=software+engineering&hl=es&source=gbs\\_navlinks\\_s](https://books.google.com/cu/books?id=2pjlAwAAQBAJ&dq=software+engineering&hl=es&source=gbs_navlinks_s).
28. SULLIVAN, W.G., WICKS, E.M. y LUXHOJ, J.T. 2004. *Ingeniería económica de DeGarmo* [en línea]. 20. México: Pearson Educación. ISBN 9702605296, 9789702605294. Disponible en: [https://books.google.com/cu/books?id=Nvh6Aag6WqsC&pg=PA329&lpg=PA329&dq=la+t%C3%A9cnica+unitaria+es+una+estimaci%C3%B3n+param%C3%A9trica?&source=bl&ots=3bJ78Zl1\\_M&sig=5-fPQuOyj0caKyDZpgh0yUbZr\\_4&hl=es-419&sa=X&ei=LNx2VdnhLs6wsASo2IDgBQ&ved=0CBwQ6AEwAA#v=onepage&q=la%20t%C3%A9cnica%20unitaria%20es%20una%20estimaci%C3%B3n%20param%C3%A9trica%3F&f=false](https://books.google.com/cu/books?id=Nvh6Aag6WqsC&pg=PA329&lpg=PA329&dq=la+t%C3%A9cnica+unitaria+es+una+estimaci%C3%B3n+param%C3%A9trica?&source=bl&ots=3bJ78Zl1_M&sig=5-fPQuOyj0caKyDZpgh0yUbZr_4&hl=es-419&sa=X&ei=LNx2VdnhLs6wsASo2IDgBQ&ved=0CBwQ6AEwAA#v=onepage&q=la%20t%C3%A9cnica%20unitaria%20es%20una%20estimaci%C3%B3n%20param%C3%A9trica%3F&f=false).
29. VILLAZÓN, Y.P., GONZÁLEZ, J.G., HERNÁNDEZ, A.V., LLANES, E.A.C., VITIER, A.G., ORAMAS, A.G. y BLANCO, Y.H. 2014. *Buenas prácticas para la migración a código abierto*. La Habana, Universidad de las Ciencias Informáticas: s.n.
30. VISUAL PARADIGM INTERNATIONAL 2015. Visual Paradigm 8.0 (formerly VP-UML 8.0) Released. *Visual Paradigm* [en línea]. [Consulta: 11 junio 2015]. Disponible en: <http://www.visual-paradigm.com/aboutus/newsreleases/vpuml80.jsp>.
31. VITIER, A.G. y GONZÁLEZ, J.G. 2011a. *Integración de los sistemas de la Plataforma de Migración a Software Libre y Código*. Ciudad de La Habana, Cuba: s.n.

---

---

## Bibliografía

1. ASHWORTH, A., HOGG, K. y HIGGS, C. 2013. *Willis's Practice and Procedure for the Quantity Surveyor* [en línea]. 13. S.l.: John Wiley & Sons. [Consulta: 5 junio 2015]. ISBN 1118589963, 9781118589960. Disponible en: [https://books.google.com/cu/books?id=nJwSZ8BeDcsC&dq=estimating+unit+method&hl=es&source=gbs\\_navlinks\\_s](https://books.google.com/cu/books?id=nJwSZ8BeDcsC&dq=estimating+unit+method&hl=es&source=gbs_navlinks_s).
2. ASHWORTH, A. y PERERA, S. 2013. *Cost Studies of Buildings* [en línea]. S.l.: Routledge. [Consulta: 5 junio 2015]. ISBN 1317902939, 9781317902935. Disponible en: [https://books.google.com/cu/books?id=iRLdAAAAQBAJ&dq=estimating+unit+method&hl=es&source=gbs\\_navlinks\\_s](https://books.google.com/cu/books?id=iRLdAAAAQBAJ&dq=estimating+unit+method&hl=es&source=gbs_navlinks_s).
3. BAKKEN, S.S., 2001. *Manual de PHP*. 2001. S.l.: s.n.
4. BANERJEE, B. 2009. *Cost accounting: Theory and Practice*. 12. New Delhi: PHI Learning.
5. BEN COLLINS-SUSSMAN 2002. Subversion Dev: Subversion 0.12.0 released. [en línea]. [Consulta: 10 junio 2015]. Disponible en: <http://svn.haxx.se/dev/archive-2002-05/0168.shtml>.
6. CENTRO DE EXCELENCIA EN COMERCIO ELECTRÓNICO 2011. 10 Programas ERP Software Libre y gratis para PYMES. *Comercio Electrónico Global* [en línea]. [Consulta: 1 junio 2015]. Disponible en: <http://www.e-global.es/erp/10-programas-erp-software-libre-y-gratis-para-pymes.html>.
7. CÉSPEDES, R.F., GARCÍA, S.P. y ROMERO, G.M.P., 2008. *Propuesta de un expediente, para los proyectos productivos del Polo de Software Libre, de la Facultad 10*. SXP. 2008. S.l.: s.n.
8. CORTÉS, G. y CASALLAS, R. 2014. Introducción a los patrones de Software. [en línea]. Universidad de los Andes. [Consulta: 12 mayo 2015]. Disponible en: <https://sistemas.uniandes.edu.co/~isis2701/dokuwiki/lib/exe/fetch.php?media=isis2701-patronesgrasp.pdf>.
9. CURTO, J.R.P. 2013. *BPM (Business Process Management): Cómo alcanzar la agilidad y eficiencia operacional a través de BPM y la empresa orientada a procesos* [en línea]. S.l.: BPMteca.com. [Consulta: 6 junio 2015]. ISBN 8461638549, 9788461638543. Disponible en: <https://books.google.com/cu/books?id=07NJBAAAQBAJ&pg=PA286&dq=m%C3%A9todos+estimaci%C3%B3n+costos&hl=es-419&sa=X&ei=nPVxVbuvDraTsQT6vr2oDA&ved=0CC0Q6AEwAw#v=onepage&q=m%C3%A9todos%20estimaci%C3%B3n%20costos&f=false>.
10. CUSCÓ, M.B.I. y WELLS, J.G., 2008. *La satisfacción del profesor de Educación Física*. 2008. S.l.: Funámbulos Editores.

11. DIOSBEL PÉREZ, YOENCY LEYVA, LIANNE REYES, ALEXANDER MARTÍNEZ y YASIEL PÉREZ 2014. *Plataforma Cubana de Migración a Código Abierto*. , vol. 8. ISSN 2227 - 1899.
12. GIBSON RESEARCH CORPORATION 2014. GRC's | TrueCrypt, the final release, archive. *Gibson Research Corporation* [en línea]. [Consulta: 11 junio 2015]. Disponible en: <https://www.grc.com/misc/truecrypt/truecrypt.htm>.
13. GRIFF, M. 2014. *Professional Accounting Essays and Assignments* [en línea]. S.l.: Miracel Griff. [Consulta: 5 junio 2015]. Disponible en: [https://books.google.com.cu/books?id=LjOpAwAAQBAJ&dq=depreciation+methods&hl=es&source=gbs\\_navlinks\\_s](https://books.google.com.cu/books?id=LjOpAwAAQBAJ&dq=depreciation+methods&hl=es&source=gbs_navlinks_s).
14. JIMÉNEZ, M.P.E.B. 2014. *Diccionario de Administración y Finanzas* [en línea]. S.l.: Palibrio. [Consulta: 5 junio 2015]. ISBN 1463354940, 9781463354947. Disponible en: [https://books.google.com.cu/books?id=8q0KBAAQBAJ&dq=definici%C3%B3n+costo&hl=es&source=gbs\\_navlinks\\_s](https://books.google.com.cu/books?id=8q0KBAAQBAJ&dq=definici%C3%B3n+costo&hl=es&source=gbs_navlinks_s).
15. JUAN, M.I., COSOLLAR, A.D., DOLZ, C.D. y ORTEGA, C.F. 2014. *Fundamentos de dirección de empresas. Conceptos y habilidades directivas* [en línea]. 2. Madrid, España: Ediciones Paraninfo, S.A. [Consulta: 11 junio 2015]. ISBN 8428399220, 9788428399227. Disponible en: <https://books.google.com.cu/books?id=X9v7CAAAQBAJ&pg=PA20&dq=eficiencia+concepto&hl=es-419&sa=X&ved=0CCsQ6AEwA2oVChMI-K28ybuGxgIVIS2MCh2wBADE#v=onepage&q=eficiencia%20concepto&f=false>.
16. LEÓN, R.A.H. y GONZÁLEZ, S.C. 2011. *El Proceso de Investigación Científica*. Ciudad de La Habana, Cuba: Editorial Universitaria. ISBN 978-959-16- 1307-3.
17. LÓPEZ, F.J.T. 2013. *Administración de proyectos de informática* [en línea]. S.l.: ECOE EDICIONES. [Consulta: 5 junio 2015]. ISBN 9586488160, 9789586488167. Disponible en: [https://books.google.com.cu/books?id=vQFaAQAAQBAJ&dq=costos+indirectos&hl=es&source=gbs\\_navlinks\\_s](https://books.google.com.cu/books?id=vQFaAQAAQBAJ&dq=costos+indirectos&hl=es&source=gbs_navlinks_s).
18. MESTRAS, J.P. 2013. *ServidoresWeb-Apache*. [en línea]. [Consulta: 23 abril 2015]. Disponible en: <http://www.fdi.ucm.es/profesor/jpavon/web/31-ServidoresWeb-Apache.pdf>.
19. MESTRAS, J.P., 2014. *Bootstrap 3.0 Aplicaciones Web/ Sistemas Web*. 2014. S.l.: s.n.
20. OPENBRAVO, 2014. *La plataforma para comercio de Openbravo. Solución flexible de comercio para minoristas ágiles* [en línea]. 2014. S.l.: s.n. [Consulta: 25 enero 2015]. Disponible en: <http://www.openbravo.com/content/pdf/openbravo-commerce-platform-brochure-es-high.pdf>.
21. ORACLE 2015. *NetBeans IDE*. *NetBeans IDE* [en línea]. [Consulta: 10 junio 2015]. Disponible en: <https://netbeans.org/features/index.html>.

22. ORTIZ, S.S., BENITEZ, A.P. y HERNÁNDEZ, D.L.G., 2015. *Estrategia de soporte técnico para el proceso de migración a software libre en Cuba*. Abril 2015. S.l.: s.n.
23. PÉREZ, D.P.Y.P., 2013. *GESPRO. Paquete para la gestión de proyectos*. 2013. S.l.: s.n.
24. PGADMIN 2015. pgAdmin 1.16 Visual Tour. *pgAdmin PostgreSQL Tools* [en línea]. [Consulta: 10 junio 2015]. Disponible en: <http://www.pgadmin.org/visualtour16.php>.
25. PISCIOTTANO, E.G.B. 2012. Implementación de OpenErp en PYME. [en línea]. [Consulta: 9 febrero 2015]. Disponible en: <file:///media/lisandra/Datos/estudio/5to/Tesis/Aplicaciones%20germanTFC0612Memoria.pdf>.
26. PMI 2013. *Guía de los fundamentos para la dirección de proyectos* [en línea]. Quinta. Estados Unidos: s.n. ISBN 978-1-62825-009-1. Disponible en: [http://portales.puj.edu.co/wjfajardo/ADMINISTRACION%20DE%20OBRAS/PRESUPUESTO/3-%20ANALISIS%20ESTRATEGICO/03%20Material%20de%20Referencia/PMBOK\\_Guide5th\\_Spanish.pdf](http://portales.puj.edu.co/wjfajardo/ADMINISTRACION%20DE%20OBRAS/PRESUPUESTO/3-%20ANALISIS%20ESTRATEGICO/03%20Material%20de%20Referencia/PMBOK_Guide5th_Spanish.pdf).
27. POTENCIER, F. y ZANINOTTO, F., 2008. *Symfony la guía definitiva*. 2008. S.l.: s.n.
28. RICH, J., MOWEN, M., HANSEN, D. y JONES, J. 2013. *Cornerstones of Financial Accounting* [en línea]. 3. S.l.: Cengage Learning. [Consulta: 5 junio 2015]. ISBN 1285607155, 9781285607153. Disponible en: [https://books.google.com/cu/books?id=jbiWAAAAQBAJ&dq=depreciation+methods&hl=es&source=gbs\\_navlinks\\_s](https://books.google.com/cu/books?id=jbiWAAAAQBAJ&dq=depreciation+methods&hl=es&source=gbs_navlinks_s).
29. ROMERO, G.M.P., 2008. *SXP. Metodología ágil para proyectos de software libre*. octubre 2008. S.l.: s.n.
30. SÁNCHEZ, R.R. 2015. Modelos de Negocio en Software Libre y los ERP: OpenBravo » Software Libre y Comunicación. *Software Libre y Comunicación* [en línea]. [Consulta: 1 junio 2015]. Disponible en: <http://ramonramon.org/blog/2015/01/14/modelos-de-negocio-en-software-libre-y-los-erp-openbravo/>.
31. SHEHORY, O. y STURM, A. (eds.) 2014. *Agent-Oriented Software Engineering: Reflections on Architectures, Methodologies, Languages, and Frameworks* [en línea]. S.l.: Springer. [Consulta: 5 junio 2015]. ISBN 3642544320, 9783642544323. Disponible en: [https://books.google.com/cu/books?id=2pjlAwAAQBAJ&dq=software+engineering&hl=es&source=gbs\\_navlinks\\_s](https://books.google.com/cu/books?id=2pjlAwAAQBAJ&dq=software+engineering&hl=es&source=gbs_navlinks_s).
32. SULLIVAN, W.G., WICKS, E.M. y LUXHOJ, J.T. 2004. *Ingeniería económica de DeGarmo* [en línea]. 20. México: Pearson Educación. ISBN 9702605296, 9789702605294. Disponible en: [https://books.google.com/cu/books?id=Nvh6Aag6WqsC&pg=PA329&lpg=PA329&dq=la+t%C3%A9cnica+unitaria+es+una+estimaci%C3%B3n+param%C3%A9trica?&source=bl&ots=3bJ78Zl1\\_M&sig=5-fPQuOyj0caKyDZpgh0yUbZr\\_4&hl=es-419&sa=X&ei=LNx2VdnhLs6wsASo2IDgBQ&ved=0CBwQ6AEwAA#v=onepage&q=la%20t%C3%A9cnica](https://books.google.com/cu/books?id=Nvh6Aag6WqsC&pg=PA329&lpg=PA329&dq=la+t%C3%A9cnica+unitaria+es+una+estimaci%C3%B3n+param%C3%A9trica?&source=bl&ots=3bJ78Zl1_M&sig=5-fPQuOyj0caKyDZpgh0yUbZr_4&hl=es-419&sa=X&ei=LNx2VdnhLs6wsASo2IDgBQ&ved=0CBwQ6AEwAA#v=onepage&q=la%20t%C3%A9cnica)

%20unitaria%20es%20una%20estimaci%C3%B3n%20param%C3%A9trica%3F&f=false.

33. VICKERSTAFF, B. y JOHAL, P. 2014. *Financial Accounting*. S.l.: Routledge.
34. VILLAZÓN, Y.P., GONZÁLEZ, J.G., HERNÁNDEZ, A.V., LLANES, E.A.C., VITIER, A.G., ORAMAS, A.G. y BLANCO, Y.H. 2014. *Buenas prácticas para la migración a código abierto*. La Habana, Universidad de las Ciencias Informáticas: s.n.
35. VITIER, A.G. y GONZÁLEZ, J.G. 2011. *Integración de los sistemas de la Plataforma de Migración a Software Libre y Código*. Ciudad de La Habana, Cuba: s.n.