

CONSTRUCCIÓN DE UN APLICATIVO PARA EL CONTROL DE ACTIVOS TECNOLÓGICOS AL INTERIOR DE UNA ORGANIZACIÓN: BENEFICIOS Y EXPERIENCIAS

CONSTRUCTION OF A SOFTWARE APPLICATION FOR MANAGEMENT AND CONTROL OF TECHNOLOGY ASSETS WITHIN A FIRM ORGANIZATION: BENEFITS AND EXPERIENCES

Adrián Alberto Jaramillo-Arango, Ing.

*Facultad de Ingenierías
Fundación Universitaria Luis Amigó
Medellín, Colombia
oviedojara@hotmail.com*

Gabriel Jaime Correa-Henao, Ph.D.

*Facultad de Ingenierías
Fundación Universitaria Luis Amigó
Medellín Colombia
gabriel.correahe@amigo.edu.co*

(Recibido el 19-03-2013. Aprobado el 10-06-2013)

Resumen. Este artículo presenta la realización de un caso de estudio que incluye el desarrollo e implementación de una herramienta de gestión de recursos tecnológicos, con aplicaciones en la mejora de los procesos de organizaciones con certificaciones en gestión de calidad. El desarrollo de la metodología y su respectiva aplicación se ha implementado en la “Fundación Universitaria Luis Amigó”, en línea con su modelo administrativo de gestión por procesos, lo cual contribuye a su reconocimiento como institución educativa de alta calidad. Los resultados de este caso de estudio han demostrado las ventajas de automatizar los diferentes procedimientos a través de herramientas de software, los cuales contribuyen a consolidar el sistema de gestión de calidad. En el artículo también se presenta una reflexión sobre la estrategia metodológica que soporta la construcción de una herramienta de software, la cual tiene en cuenta la combinación de los paradigmas: ingeniería de requerimientos y los enfoques administrativos propios de la organización, para facilitar la gestión del conocimiento y la administración de activos tecnológicos.

Palabras clave: Gestión del Conocimiento, Ingeniería de Requerimientos, Programación de Objetos, PHP.

Abstract. This paper shows the results of a case-study which has been applied to both development and implementation of a software tool for management of technologic assets. The development of the methodology and its respective application has been implemented in “Fundación Universitaria Luis Amigó”, according to its administrative structure model that aims to its recognition as high quality educational institution. The results of this study have demonstrated the advantages of automating various processes by means of software tools, which help to strengthen the quality management system. The article also shows a discussion on the strategy that supports the construction each organization, in order to facilitate knowledge management and asset management technology.

Keywords: Knowledge Management, Software Engineering, Oriented Objects Programming, PHP.

1. INTRODUCCIÓN

Desde los primeros años del siglo XXI se hace evidente, para las organizaciones o entidades con cualquier tipo de razón social, la necesidad de manejar su información mediante la gestión de procesos que faciliten su competitividad. Dicha necesidad exige el uso de herramientas tecnológicas que le permita a tales entidades estar a la vanguardia de la gestión de sus procesos. Esta continua evolución tecnológica exige a las organizaciones, alinear sus procesos administrativos con la gestión tecnológica, facilitando muchas de las tareas inherentes a los diferentes procesos y funciones, lo cual representará optimizaciones en costos y en tiempos [1], [2], [3].

El modelo empresarial surgido desde principios del siglo XX, y que está plasmado en la teoría de la administración, plantea la necesidad de implementar sistemas de gestión que permitan controlar, de forma sistemática, las actividades y procesos de la empresa, con la participación e implicación de todos sus trabajadores, para lograr los resultados previstos [4], [5]. En estos modelos de gestión no basta con tener en cuenta solamente los parámetros económicos y de productividad, puesto que la satisfacción de los trabajadores, de los clientes y del entorno social en el que desarrolla su actividad, las actuales exigencias en materia de seguridad y salud en el trabajo, de la calidad y medioambientales, deben ocupar posiciones preponderantes y de mejora continua.

Los directivos que pretendan mantener sus organizaciones en el mercado, tendrán que adaptarse a las anteriores exigencias e implantar sistemas de gestión informatizados y autogestionados que permitan que sus productos, procesos o servicios tengan elementos cualitativos que sean de peso, den confianza y favorezcan la decisión de compra por los clientes [6], [7].

En este artículo se pretende mostrar cómo el capital humano de las organizaciones es el principal gestor de evolución en las mismas; dicho capital es el principal generador de conocimiento que, a través de aportes significativos, puede contribuir al éxito de las empresas.

Se realiza un caso de estudio que pretende presentar un proceso de mejora al interior de la Fundación Universitaria Luis Amigó, específicamente un procedimiento del departamento de infraestructura tecnológica donde, a la fecha (año 2014), existen

procedimientos muy básicos para el control de los elementos de cómputo [7]. Dichos procedimientos se fundamentan en la escritura de documentos, lo cual es poco productivo e ineficaz. La FUNLAM desea mejorar sus procesos en virtud de las políticas de calidad.

Por tal razón se describirán métodos enfocados desde el punto de vista administrativo y de ingeniería de software, con el fin de desarrollar una herramienta informática que automatice el proceso de inventario de activos, y el control y parametrización de funcionalidades inherentes al proceso.

El presente trabajo dará cuenta del desarrollo de esta herramienta de gestión, la cual se ha denominado SPARTEC (Sistema Para la Administración de Recursos Tecnológicos). La primera parte del artículo presenta aspectos relevantes, desde el enfoque administrativo, incluyendo algunas definiciones en torno al paradigma de gestión por procesos, así como las estrategias de mejoramiento continuo en una organización. La segunda parte de este artículo presenta aspectos relacionados con el desarrollo técnico y la ingeniería de software. Lo anterior incluye la descripción de los pasos necesarios en el desarrollo de una herramienta informática, desde el levantamiento de requisitos, análisis y modelos de datos. La última parte presenta los beneficios en la aplicación y uso de la herramienta, en un marco de gestión de un sistema de calidad a nivel institucional.

2. METODOLOGÍA DE DESARROLLO PARA UN CASO DE ESTUDIO

El término “sociedad del conocimiento” hace referencia a todo aquello que permite, al ser humano, realizarse y contribuir al desarrollo de su sociedad [8]. Por este motivo, y gracias al aporte de personas idóneas en el ámbito tecnológico, ha sido posible encontrar mecanismos que impulsen a la humanidad a desarrollar estrategias y métodos que ayudan a resolver los grandes problemas y enigmas que propone la realidad actual [1]. En este contexto surge la necesidad de implementar una herramienta informática que permita contribuir al mejoramiento de procesos a nivel organizacional en la Funlam. Lo anterior requiere la formación de sinergias entre los componentes de software, desde el punto de vista lógico (Ingeniería de sistemas de información), y las diferentes modalidades de alcanzar la calidad institucional [8].

En un marco de ingeniería informática existen metodologías y herramientas que facilitan el desarrollo tecnológico de la sociedad [6], los cuales se describen en esta sección. Especial énfasis tiene el desarrollo de sistemas bajo el paradigma de programación orientada a objetos, la cual se populariza cada vez más, dado que los lenguajes fundamentados en dicho paradigma permiten desarrollar software a partir del análisis y diseño de soluciones, con criterios de adaptabilidad, robustez, fiabilidad y calidad [10], [11], [12].

De esta manera, estas herramientas de software contribuyen a la mejora de procesos y no se convierten en una carga para los procesos donde se pretende brindar soluciones [1], [5]. La Ingeniería de Software reconoce que una buena estrategia para aumentar la calidad de sus productos requiere de una mejora de procesos en las organizaciones [13]; por ejemplo, a través del soporte tecnológico por medio de herramientas groupware [5], [14], [15], [16], las cuales ayudan a gestionar los proyectos de mejora de procesos.

En esta sección se presenta la aplicación de una metodología de desarrollo para crear una herramienta de software que pretende dar solución a la problemática que enfrenta el área de tecnologías de la Fundación Universitaria Luis Amigó. Dicha herramienta se desarrolló bajo estándares y normas de la Ingeniería de Sistemas, con aplicaciones a la ingeniería de procesos; estos procesos serán abordados desde conceptos globales que se expondrán en los apartados siguientes.

2.1 Antecedentes administrativos

El diseño e implementación del Sistema de Gestión Institucional –SIGI– para la Fundación Universitaria Luis Amigó –Funlam–, basado en la Norma NTC ISO 9001 [19], [20], tiene como propósito proveer a la institución de una herramienta de gestión establecida en procesos que le permita tener un mejor control de sus actividades, el logro de sus objetivos —de una manera eficaz y eficiente—, y adoptar la cultura del mejoramiento continuo para el alcance de altos niveles de competitividad [16].

El SIGI proporciona una descripción detallada de los procesos y procedimientos, lo cual se constituye en el marco de referencia de las actividades, de los responsables, de los insumos y productos relaciona-

dos a cada uno de los procesos, suministrando un conocimiento claro de cómo se están desarrollando las diferentes labores a su interior [7]. La herramienta proporciona información a la alta dirección, a los clientes, a proveedores, al personal administrativo y demás interesados, lo que demuestra el compromiso de la Funlam con el mejoramiento de la calidad de todos sus servicios.

De igual manera, el SIGI constituye la plataforma para soportar la generación de indicadores de medición en el cumplimiento de metas de la organización. También permite definir los mecanismos e instrumentos necesarios para recopilar la información hacia la toma de decisiones operativas, tácticas y estratégicas de la Funlam, de acuerdo a los sistemas de gestión con indicadores clave [10].

En coherencia con lo anterior, el Sistema de Gestión Institucional de la Funlam se propone armonizar los requisitos de la Norma ISO 9001, que son genéricos, con los requisitos específicos del sector de la educación superior, en lo concerniente a los procesos de autoevaluación y acreditación institucionales y de programas [20]. Los procesos de certificación y acreditación de la educación superior, como acciones permanentes de mejora continua y aseguramiento de la calidad, determinan una dinámica que moviliza la estructura organizacional de las instituciones de educación superior hacia la consecución de los objetivos, plasmados en sus postulados estratégicos. Se hace evidente que todo desarrollo de proyectos debe tener claros los requerimientos en cuanto a cuál va a ser su forma de trabajar, es por esto que para el presente trabajo se tienen en cuenta diferentes elementos que, directa o indirectamente, se involucran y participan activamente del éxito del proyecto [3], [21].

En 2013, el proceso de administración de los recursos tecnológicos de la Fundación Universitaria Luis Amigó se enmarcó dentro del proceso de activos fijos de la institución; es decir, el inventario de activos lo controla el departamento de contabilidad de la institución, pero lo hace únicamente en modo de valor económico, es por esta razón que el departamento de infraestructura ingresa en esta problemática, ya que a parte de la necesidad de controlar los bienes institucionales, también es necesario controlar todo lo que tiene que ver con gestión de requerimientos, control de asignaciones, control de reparaciones, control de equipos obsoletos, control de ubicaciones, generación de órdenes de salida a reparación, rea-

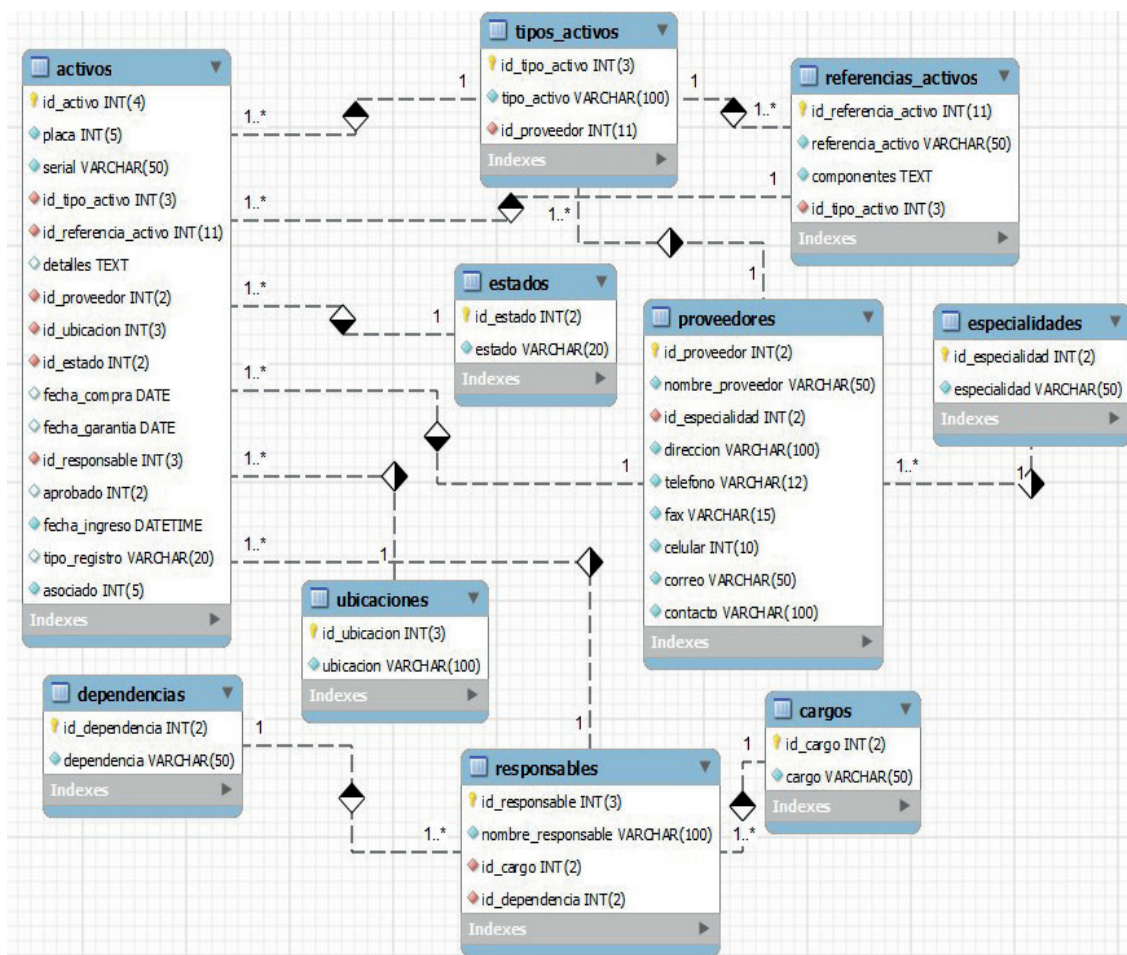


Fig. 1. Modelo de base de datos entidad-relación para administración de recursos tecnológicos en la Funlam.

lización automática de las respectivas actas de dar de baja, gestión de elementos disponibles, gestión histórica de movimientos; es decir, una tarea un poco más allá de un simple control económico que permite reforzar la idea de mejoramiento continuo, al que la universidad apunta [22].

2.2 Definición del modelo de datos

El desarrollo de la herramienta informática requiere la configuración de una base de datos correspondiente a una agrupación de datos que, interrelacionados, conforman una ideología única que, posteriormente, se trasformarán en información valiosa para la organización, permitiendo así la correcta toma de decisiones con respecto a una situación específica [23].

Para el caso de estudio que se presenta en esta sección, se propone utilizar el modelo de la base de datos entidad-relación, el cual se obtiene en la fase de

diseño de la herramienta de software, originalmente propuesto por Chen [23] y que data de la década de 1970. Dicho modelo se caracteriza por utilizar una serie de símbolos y reglas para representar los datos y sus relaciones [30]. El modelo de base de datos entidad-relación permite representar, de manera gráfica, la estructura lógica de un software [24]. La Figura 1 presenta la concepción de dicho modelo para el caso de estudio, objeto de este artículo en la Funlam, la cual constituye el pilar fundamental de la aplicación.

El modelo de base de datos se compone de diferentes elementos que permiten determinar el rumbo, la factibilidad y la calidad con la que contará la herramienta a construirse [16]. Para el estudio del caso que se presenta en este artículo, el modelo entidad-relación se aplica en los procesos de administración de recursos tecnológicos, dentro de un marco de cumplimiento de la política de calidad institucional.

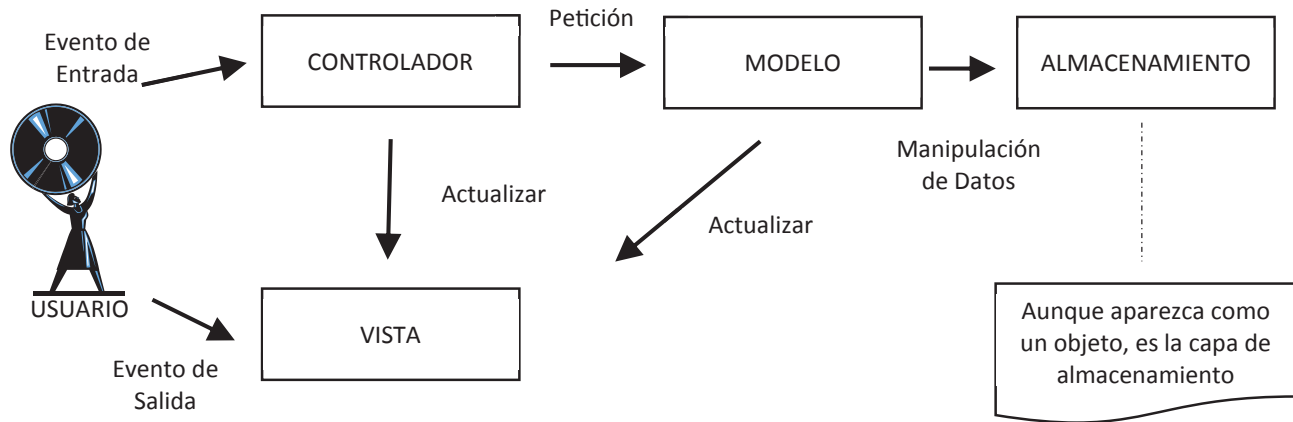


Fig. 2. Concepción de la Arquitectura Informática bajo el modelo Smalltalk [26].

2.3 Desarrollo de la Arquitectura Informática

Para el desarrollo de la herramienta informática se utilizan patrones establecidos en los procesos del modelo de datos, presentados previamente en la Fig. 1. Consecuentemente, la herramienta a implementar se despliega bajo una metodología de desarrollo ágil, la cual permite ejecutar modificaciones rápidas sin afectar otras componentes del sistema. Esto garantiza un nivel de robustez para afianzarse como aplicativo que proporciona confiabilidad en la toma de decisiones a nivel organizacional.

Para el caso de estudio se propone trabajar con el patrón MVC (Patrón Modelo Vista Controlador), correspondiente a una arquitectura de diseño propuesta por Trygve Reenskaug [28], cuyos principios se tienen en cuenta en la concepción de la arquitectura bajo el modelo Smalltalk [26], consistente en subdividir la aplicación en tres módulos, según se aprecia en la Fig. 2.

De la Fig. 2 se pueden apreciar las siguientes subdivisiones, en la concepción de la arquitectura informática para la construcción de la herramienta [28], [26]:

1. La vista del usuario (la interfaz a la que accede el usuario).
2. La lógica de control para captar los eventos que el usuario ha generado a través de la interfaz.
3. El modelo que gestiona los datos según le indique la lógica de control.

El flujo que sigue el patrón de la arquitectura informática generalmente es el siguiente [25]:

1. El usuario interactúa con la interfaz de usuario, por ejemplo al pulsa un enlace. Este punto corresponde, en el patrón, a la vista.
2. El controlador recibe (a través de la interfaz) la notificación de la acción solicitada por el usuario; es decir, el controlador gestiona el evento que llega desde la vista, producido por un usuario.
3. El controlador accede al modelo, ya sea con el fin de consultar datos o actualizarlos, posiblemente modificándolo de forma adecuada a la acción solicitada por el usuario.
4. El controlador delega, a los objetos de la vista, la tarea de desplegar la interfaz de usuario. La vista obtiene sus datos del modelo para generar la interfaz apropiada para el usuario, donde se refleja los cambios en el modelo (por ejemplo, produce un listado de las películas que tal usuario tiene). El modelo no debe poseer conocimiento directo sobre la vista. Por lo general, el controlador no pasa objetos de dominio (el modelo) a la vista, aunque puede dar la orden a la vista para que se actualice; sin embargo, en algunas implementaciones, la vista no tiene acceso directo al modelo, dejando que el controlador envíe los datos del modelo a la vista.
5. La interfaz de usuario espera nuevas interacciones del usuario, comenzando el ciclo nuevamente.

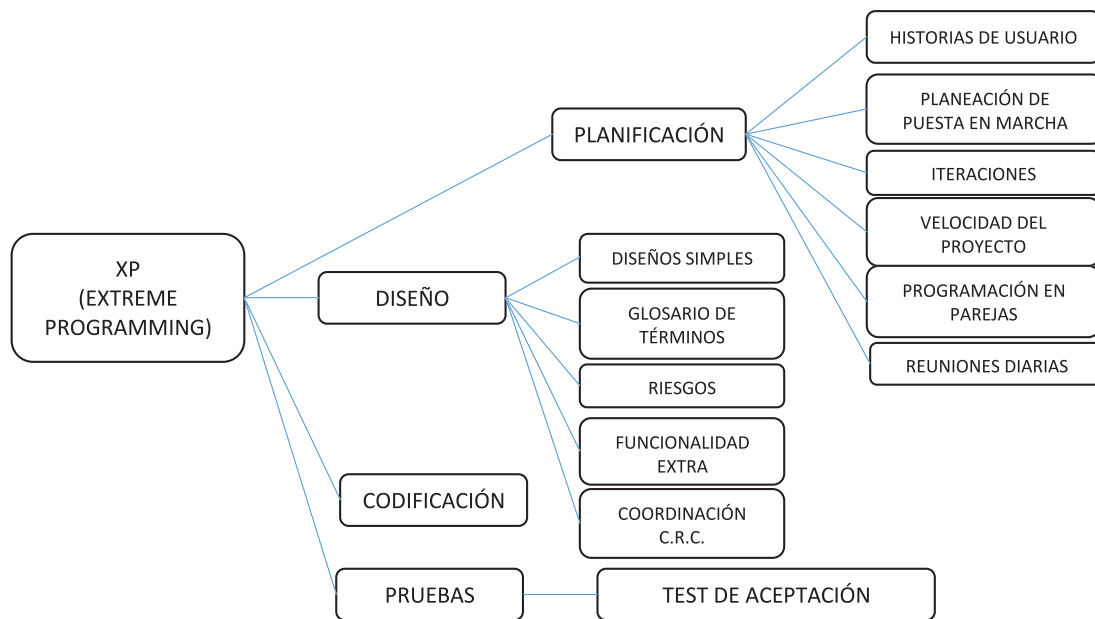


Fig. 3. Estructura de la metodología rápida XP (Extreme Programming) [27] [34].

2.4 Metodología de desarrollo rápido

Existen numerosas propuestas metodológicas que inciden en distintas dimensiones del proceso de desarrollo. Por una parte se da crédito a aquellas propuestas más tradicionales que se centran especialmente en el control del proceso, estableciendo rigurosamente las actividades involucradas, los artefactos que se deben producir y las herramientas y notaciones que se usarán [27].

Estas propuestas han demostrado ser efectivas y necesarias en un gran número de proyectos, pero también han presentado problemas en otros. Una posible mejora consiste en incluir el desarrollo de actividades, artefactos y restricciones, de un modo más amplio, basándose en los puntos débiles detectados, comparado con aquellas situaciones en las que no se utilizan metodologías formales [31].

Teniendo en cuenta lo anterior se opta por trabajar con la metodología ágil XP (Extreme Programming) [25], [27]. Dicha metodología XP se caracteriza porque da mayor valor al cliente, a la colaboración con el cliente y al desarrollo incremental del software con iteraciones muy cortas. Este enfoque está mostrando su efectividad en proyectos con requisitos muy cambiantes y cuando se exige reducir drásticamente los tiempos de desarrollo, al tiempo que garantiza alta calidad en la concepción de la arquitectura de software [34], [25]. La Fig. 3 presenta la estructura

general que conforma la metodología de desarrollo rápido. También especifica los elementos que componen el orden a manejar en cada una de las fases de la metodología.

2.4.1 Recolección de información

La trazabilidad de requisitos consiste en la especificación y validación de los servicios que debe proporcionar el sistema, así como las restricciones sobre las que se deberá operar. Es un proceso iterativo y cooperativo de análisis del problema, documentando los resultados en una variedad de formatos y probando la exactitud del conocimiento adquirido, de tal forma que el aplicativo construido cumpla con los niveles de calidad exigidos por el cliente [17], [29].

Para el estudio del caso documentado en este artículo, se tomaron diferentes mecanismos de recolección de información entre los clientes del proceso, incluyendo a funcionarios de la Funlam, como son los cargos de jefe del departamento de sistemas y profesionales y asistentes del área funcional de infraestructuras informáticas. La técnica de recolección de información se fundamentó en entrevistas que permitieran conocer las necesidades y expectativas de los usuarios en la herramienta informática.

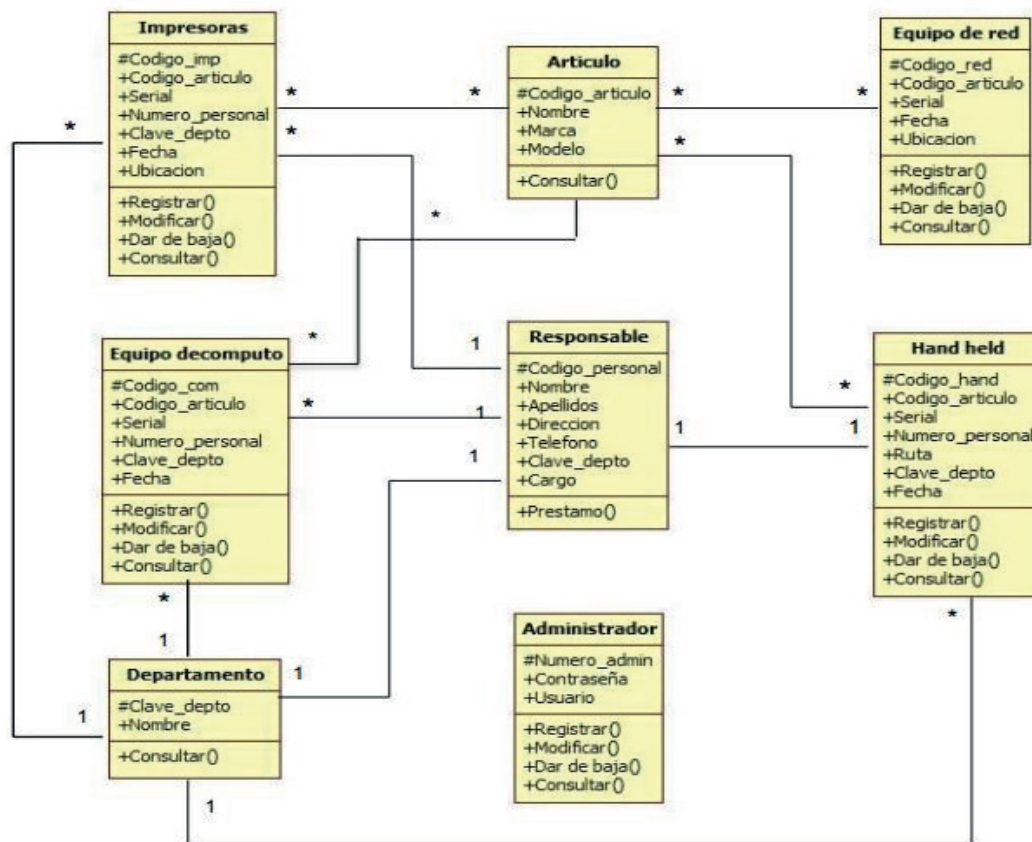


Fig. 4. Aparte del diagrama de clases para la herramienta de administración de recursos tecnológicos en la Funlam.

En el proceso de recolección de información se identificó la necesidad de cumplir con el Plan Informático Institucional [7], el cual exige que las herramientas informáticas cumplan con parámetros de arquitectura monolítica, en configuración cliente-servidor o en configuración por capas; esta última corresponde a la arquitectura elegida para el caso de estudio.

Dichos parámetros se formalizan con un lenguaje estándar y unificado, es decir, todo el software desde su concepción, análisis, desarrollo y pruebas está documentado de la mejor forma, siguiendo estándares y procedimientos según normas internacionales o las propias de cada organización [30]. En el caso de estudio se eligió un lenguaje gráfico, que permite especificar y documentar un sistema de software de modo estándar (incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema).

En consecuencia, se ha verificado la posibilidad de utilizar el lenguaje UML 2.0 que incorpora el manejo de elementos netamente estructurales e igualmente cuenta con una notación estándar y semánticas

esenciales (aunque abstractas) para el modelado de un sistema orientado a objetos [25], [32], [33].

2.4.2 Diagrama de clases

Un diagrama de clases se refiere a la representación estática que describe la estructura de un sistema en cuanto a la relación de sus componentes internos [32]. La Fig. 4 representa un aparte del diseño del diagrama de clases, aplicado al caso de estudio, de un sistema informático para administración de activos.

2.4.3 Elección del lenguaje de programación orientado a objetos

El desarrollo de aplicaciones de software incluye la elección del lenguaje de programación [30], el cual se refiere al conjunto de códigos para que la máquina pueda interpretar las acciones y solicitudes de las personas u otros elementos que interactúan con la herramienta informática.

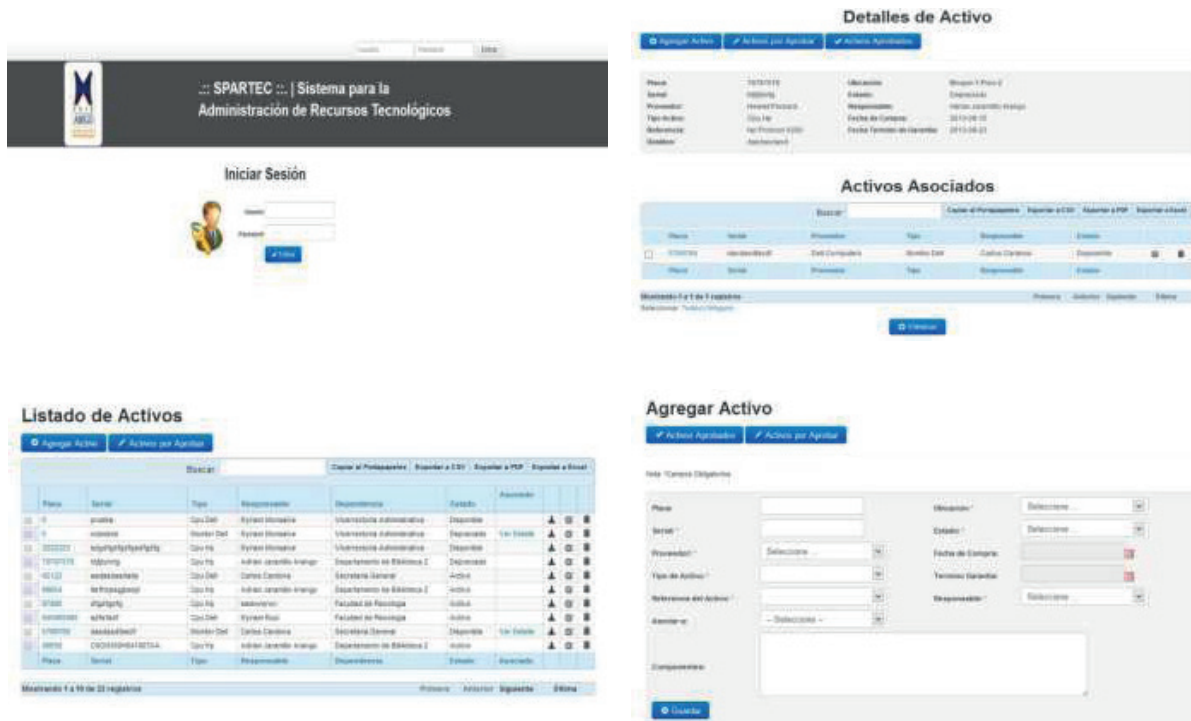


Fig. 5. Presentación de la funcionalidad del sistema implementado en la Funlam.

Teniendo en cuenta la amplia difusión del paradigma de programación orientado a objetos [32], para este caso de estudio se prefiere utilizar un lenguaje fundamentado en dicho paradigma, que sea de libre acceso. La construcción de la herramienta informática documentada, en este caso, se fundamentó en el uso de la plataforma PHP [18], [34], que es un lenguaje de programación que permite técnicas de orientación a objetos y está enfocado a la plataforma web.

3. PROTOTIPO Y RESULTADOS

El caso de estudio presentado en este artículo ha derivado en la realización de la herramienta SPARTEC (Sistema Para la Administración de Recursos Tecnológicos de la Funlam). Esta sección proporciona algunos resultados y beneficios que derivan de aplicar la herramienta de software, que responde a las necesidades del departamento de infraestructura tecnológica de la Funlam, al tiempo que contribuye al cumplimiento de objetivos de calidad a nivel institucional [7].

3.1 Desarrollo de la herramienta de software

El sistema SPARTEC se fundamenta en un patrón MVC que facilita, en gran medida, el control de la aplicación en lo que se refiere al desarrollo [28].

Dicha aplicación informática demanda el establecimiento de sesiones para sus usuarios, requiriendo una identificación mediante contraseña, para dar acceso a los diferentes módulos que componen el sistema. La herramienta SPARTEC administra el control de acceso, mediante el algoritmo de encriptamiento en la contraseña con *Hash Message-Digest Algorithm* (MD5) [32], [34].

Una vez los usuarios ingresen correctamente al sistema, se encontrarán con una serie de módulos que, en conjunto, conforman la base funcional del mismo. En el menú de la parte superior se encuentran los módulos más importantes, y en el menú lateral se encuentran las opciones de apoyo a los módulos globales.

La Fig. 5 presenta algunas funcionalidades específicas del sistema, incluyendo algunos de los principales módulos de la herramienta SPARTEC, correspondientes al módulo de control y gestión de activos tecnológicos, el de control y gestión de software y el módulo de telefonía.

Listado de Ordenes de Salida

Buscar:		Copiar al Portapapeles	Exportar a CSV	Exportar a PDF	Exportar a Excel			
ID	Placa	Serial	Tipo Activo	Motivo Salida	Fecha			
<input type="checkbox"/>	18	0	prueba	Cpu Dell	Mantenimiento Preventivo	2013-10-23	Ver	
<input type="checkbox"/>	17	0	prueba	Cpu Dell	Mantenimiento Preventivo	2013-10-09	Ver	
<input type="checkbox"/>	16	0	prueba	Cpu Dell	Mantenimiento Preventivo	2013-09-05	Ver	
<input type="checkbox"/>	15	0	prueba	Cpu Dell	Mantenimiento Preventivo	2013-08-16	Ver	
<input type="checkbox"/>	14	98565	asfasfaf	Cpu Dell	Mantenimiento Preventivo	2013-08-21	Ver	
<input type="checkbox"/>	13	98565	asfasfaf	Cpu Dell	Mantenimiento Correctivo	2013-07-11	Ver	
<input type="checkbox"/>	12	98565	asfasfaf	Cpu Dell	Mantenimiento Correctivo	2013-07-16	Ver	
<input type="checkbox"/>	11	98565	asfasfaf	Cpu Dell	Mantenimiento Preventivo	2013-07-23	Ver	
<input type="checkbox"/>	10	98565	asfasfaf	Cpu Dell	Mantenimiento Correctivo	2013-07-06	Ver	
<input type="checkbox"/>	9	9000	pruebatraza	Cpu Hp	Préstamo fuera de las Instalaciones	2013-07-16	Ver	
ID	Placa	Serial	Tipo Activo	Motivo Salida	Fecha			

Mostrando 1 a 10 de 13 registros

Primera Anterior Siguiente Última

Fig. 6. Ejemplo de los reportes que genera el sistema implementado.

Para el apartado de activos se han tomado conceptos sencillos relacionados con el modelo CRUD (*Create, Read, Update y Delete*) [35], que permite hacer de forma eficiente el trabajo del administrador; por consiguiente, se cuentan con métodos para listar la información contenida en la base de datos.

La herramienta SPARTEC también cuenta con formularios que permiten ingresar información nueva, actualizar la información existente, así como integrar un algoritmo de aprobación automática y de envío de correos, cada vez que sucede una novedad dentro del sistema.

Cabe mencionar que el software trabaja con la implementación de diferentes *plugins* que permiten ejecutar los procesos de desarrollo de una manera sencilla para el usuario. Las vistas para la aplicación se trabajan con el gestor de plantillas *Smarty* [5], [32] y la interacción entre los diferentes elementos (como los de búsquedas y validaciones) se trabaja con el plugin *jqueryUI* [12], [32].

3.2 Apoyo a la toma de decisiones

El sistema que se desarrolló fue pensado para proporcionar apoyo a la toma de decisiones en el departamento de infraestructura tecnológica; por esta razón se hizo necesario desarrollar una serie de procedimientos que se encargaran de la generación de

algunos reportes. La Fig. 6 presenta un prototipo de informe que realiza el sistema después de ejecutar sus módulos internos.

Este tipo de informes permiten apoyar la toma de decisiones, dado que proporcionan listados sobre el historial de órdenes de salida para los elementos informáticos que entran y salen de la institución. También presentan estadísticas detalladas de las razones.

El hecho de incluir un sistema que permita exportar en cualquier momento los posibles listados de cualquier módulo, facilita en gran manera la gestión de los procesos a nivel del departamento, ya que agiliza y facilita muchos de los procesos que anteriormente se hacían de forma manual.

4. CONCLUSIONES

En este artículo se han presentado los pasos para realizar un aplicativo que puede ser usado para el control y administración de recursos a través de una herramienta de software. Este caso de estudio ha tenido en cuenta la revisión e implementación de modelos y buenas prácticas, la identificación de las tendencias de gestión de activos en la industria y el estudio de herramientas de desarrollo disponibles. Lo anterior ha permitido caracterizar los requisitos de

una herramienta de software para apoyar la gestión de administración de recursos tecnológicos.

Ha sido posible identificar fortalezas, debilidades y oportunidades de mejoramiento con el ánimo de participar activamente del progreso continuo requerido en el sistema de calidad de la Funlam. Un diagnóstico inicial presentaba dificultades en los mecanismos de control institucionales, debido a la carencia de métodos cualitativos y cuantitativos, pues se confiaba sólo en el uso de herramientas poco automatizadas (como el paquete de ofimática de Microsoft) para llevar el control del inventario y el control de elementos específicos que se manejan al interior del área.

Este caso de estudio demostró la conveniencia de poner en funcionamiento los proyectos de ingeniería de software que contribuyan al mejoramiento de los procesos. Un trabajo futuro permitirá impulsar el uso de esta aplicación en otras áreas institucionales, teniendo en cuenta el impacto positivo dentro del sistema de gestión de calidad en la organización.

REFERENCIAS BIBLIOGRÁFICAS

- [1] J. A. Fraguera Formoso, et al., "La integración de los sistemas de gestión: necesidad de una nueva cultura empresarial". *Revista Dyna – Facultad de Minas, Medellín, Colombia*, vol. 78, no. 167, pp. 44-49, Abril 2011. Disponible en: http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=S0012-73532011000300005&lng=en&nrm=iso
- [2] L. Pedraja-Rejas, E. Rodríguez-Ponce, J. Rodríguez-Ponce., "Sociedad del conocimiento y dirección estratégica: Una propuesta integradora". *Revista Asociación Interciencia*, vol. 31, no. 8, pp. 570-576, 2006. ISSN 0378-1844. Disponible en: http://www.scielo.org.ve/scielo.php?script=sci_arttext&pid=S0378-18442006000800006&lng=es&nrm=iso
- [3] Project Management Institute., "A guide to the project management body of knowledge (PMBOK Guide)", 2008. PMI. Disponible en: <http://www.worldcat.org/isbn/9781933890517>
- [4] L. Rodríguez-Martínez, M. Mora, F. Álvarez, L. Garza, H. Durán, and J. Muñoz. "Review of Relevant System Development Life Cycles (SDLCs) in Service-Oriented Software Engineering (SoSE)," *Journal of applied research and technology*, vol. 10, no. 2, pp. 94-113, 2012. Disponible en: http://www.scielo.org.mx/scielo.php?script=sci_arttext&pid=S1665-64232012000200003&lng=es&tlng=en
- [5] A. Bedate, , L. C. Herrero, J. A. Sanz, "Economic valuation of the cultural heritage: application to four case studies in Spain", *Journal of Cultural Heritage*, vol. 5, no 1, pp. 101-111, 2004. ISSN 1296-2074. Disponible en: <http://dx.doi.org/10.1016/j.culher.2003.04.002> .
- [6] R. M. Stair, G. W. Reynolds, *Principios de sistemas de información: enfoque administrativo*, 4th ed., 2000, Ed. Thomson.
- [7] A. Mendes da Silva, "Capítulo 1 - Panorâmica da Programação Orientada a Objetos, In *Introdução À Programação orientada a objetos Com C++*", Edited by Antonio Mendes da Silva Filho, Elsevier Editora, 2010, pp. 1-13, ISBN 9788535237023. Disponible en: <http://www.sciencedirect.com/science/article/pii/B9788535237023500017>
- [8] Carballo R., "Innovación y gestión del conocimiento: modelo, metodología, sistemas y herramientas de innovación", 2006. Ed. Díaz de Santos.
- [9] Nuo Li, Tao Xie, Maozhong Jin, Chao Liu, "Perturbation-based user-input-validation testing of web applications", *Journal of Systems and Software*, vol. 83, no. 11, pp. 2263-2274, 2010. ISSN 0164-1212. Disponible en: <http://dx.doi.org/10.1016/j.jss.2010.07.007>
- [10] Fundación Universitaria Luis Amigó. *Manual de Calidad*, Fondo Editorial Funlam, 2013. Disponible en: [http://www.funlam.edu.co/uploads/gestioncalidad/447_OD-001_-_Manual_de_Calidad_\(V4\).pdf](http://www.funlam.edu.co/uploads/gestioncalidad/447_OD-001_-_Manual_de_Calidad_(V4).pdf)
- [11] C. Macías-Gelabert, A. Aguilera-Martínez., "Contribución de la gestión de recursos humanos a la gestión del conocimiento". *Revista de Estudios Gerenciales*, vol. 28, no. 123, pp. 133-148, 2012. ISSN 0123-5923. Disponible en: http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=S0123-59232012000200009&lng=en&nrm=iso
- [12] Y. Gorbaneff, S. Torres, J. F. Cardona, "El concepto de incentivo en administración. Una revisión de la literatura". *Revista Economía Institucional*, vol. 11, no. 21, pp. 73-91, 2009. ISSN 0124-5996. Disponible en: http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=S0124-59962009000200006&lng=en&nrm=iso
- [13] Pires, Machado, "Gestión por Procesos en el Diseño de las Organizaciones". *Información Tecnológica*, vol. 17, no. 1, pp. 35-44, 2006. ISSN 0718-0764. Disponible en: <http://dx.doi.org/10.4067/S0718-07642006000100005>
- [14] J. Canós, P Letelier, M. C. Penadés, "Metodologías Ágiles en el desarrollo de Software," Universidad Politécnica de Valencia, Valencia, 2003. Disponible en: <http://www.willydev.net/descargas/masyxp.pdf>

- [15] P. Díaz, Sánchez, «Metamorfosis: un marco para el análisis de requisitos funcionales», *Workshop em Engenharia de Requisitos*, pp. 233-244, 2005. Disponible en: <http://www.lbd.dcc.ufmg.br/bdbcomp/servlet/Trabalho?id=4531>
- [16] W. L. Pantoja, C. A. Collazos, V. M. R. Penichet, «Entorno colaborativo de apoyo a la mejora de procesos de software en pequeñas organizaciones de software». *Revista Dyna*, vol. 80, no. 177, pp. 40-48, Febrero 2013. Disponible en: http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=S0012-73532013000100005&lng=en&nrm=iso
- [17] M. J. Escalona, N. Koch, «Ingeniería de Requisitos en Aplicaciones para la Web: Un estudio comparativo», Universidad de Sevilla, 2002. Disponible en: <http://www.lsi.us.es/docs/informes/LSI-2002-4.pdf>
- [18] E. Velásquez-Rodríguez, Á. Custodio-Ruiz, «Sistema para la gestión del mantenimiento para un control supervisorio basado en software libre con PHP». *Universidad, Ciencia y Tecnología*, vol.15, no. 59, pp. 103-113, 2011. ISSN 1316-4821. Disponible en: http://www.scielo.org.ve/scielo.php?script=sci_arttext&pid=S1316-48212011000200007&lng=es&nrm=iso
- [19] E. Campos, Teixeira, C. Lima, «Adotando a tecnologia de informação: análise da implementação de sistemas de "groupware"». *RAE electronica*, vol. 3, no. 1, pp. 0-0, 2004. ISSN 1676-5648. Disponible en: <http://dx.doi.org/10.1590/S1676-56482004000100006>
- [20] M. Suárez-Pita, «Las normas ISO 9000: 2000 en las actividades de registro sanitario». *Revista Cubana Higiene Epidemiológica*, vol. 41, no. 1, pp. 0-0, 2003. Disponible en: http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1561-30032003000100010&lng=es&nrm=iso ISSN 1561-3003
- [21] L. Solarte-Pazos, L. F. Sánchez-Arias, «Gerencia de proyectos y estrategia organizacional: el modelo de madurez en Gestión de Proyectos CP3M-V5.0». *Revista Innovar*, vol. 24, no 52, pp. 5-18, 2014. ISSN 0121-5051. Disponible en: http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=S0121-50512014000200002&lng=en&nrm=iso
- [22] Ó. Parada-Gutiérrez, «Un enfoque multicriterio para la [11] toma de decisiones en la gestión de inventarios». *Cuaderno en Administración*, vol. 22, no. 38, pp. 169-187, 2009. ISSN 0120-3592. Disponible en: http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=S0120-35922009000100009&lng=en&nrm=iso
- [23] I. Jacobson, G. Booch, J. Rumbaugh, «El proceso unificado de desarrollo de software», Addison Wesley Reading, vol. 7, 2000.
- [24] P. P. Shan Chen, «The entity-relationship model—toward a unified view of data», *ACM Transactions on Database Systems (TODS)*, vol. 1, pp. 9-36, 1976.
- [25] J. Joskowicz, «Reglas y prácticas en eXtreme Programming». Universidad de Vigo, 2008, p. 22. Disponible en: <http://iie.fing.edu.uy/~josej/docs/XP%20-%20Jose%20Joskowicz.pdf>
- [26] A. Goldberg, «Smalltalk-80: The Interactive Programming Environment», Addison-Wesley series in computer science. p. 516, 1983. ISBN 100201113724.
- [27] *Extreme Programming: A gentle introduction*, 2014 [online]. Disponible en <http://www.extremeprogramming.org/>.
- [28] T. Reenskaug, P. Wold, and O. Arild-Lehne. «Working With Objects: The OOram Software Engineering Method», Prentice Hall, 1995. ISBN 978-0-13-452930-1.
- [29] G. Booch, J. Rumbaugh, I. Jacobson, «UML: el lenguaje unificado de modelado», Addison-Wesley Iberoamericana España, 1999.
- [30] R. Díaz, «Las metodologías ágiles como garantía de calidad del software», *REICIS Revista Española de Innovación, Calidad e Ingeniería del Software*, vol. 5, pp. 40-43, 2009.
- [31] C. M. Zapata-Jaramillo, F. Arango-Isaza, «The UML method: a problem-based software development method», *Ingeniería e Investigación*, vol. 29 no. 1, pp. 69-75, Abril 2009.
- [32] A. Hernández-González, «Un Método para el Diseño de la Base de Datos a partir del Modelo Orientado a Objetos». *Computación y Sistemas*, vol.7, no. 4, pp. 224-238, 2004. ISSN 1405-5546. Disponible en: http://www.scielo.org.mx/scielo.php?script=sci_arttext&pid=S1405-55462004000200002&lng=es&nrm=iso
- [33] F. Asteasuain, A. Javed, «Applying aspect oriented technology to relational data bases: The replication case». *Ingeniería y Desarrollo*, no. 25, pp. 218-234, 2009. ISSN 2145-9371. Disponible en: http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=S0122-34612009000100012&lng=en&nrm=iso
- [34] M. Oliveira, M. Caldeira, «Um «framework» para a gestão do conhecimento nas organizações». *Revista Portuguesa e Brasileira de Gestão*. vol.7, no. 1, pp. 33-43, 2008, ISSN 1645-4464. Disponible en: http://www.scielo.gpeari.mctes.pt/scielo.php?script=sci_arttext&pid=S1645-44642008000100005&lng=pt&nrm=iso
- [35] M. Heller, 2007. «REST and CRUD: the Impedance Mismatch», Developer World. InfoWorld. [online]. Disponible en: <http://www.infoworld.com/d/developer-world/rest-and-crud-impedance-mismatch-927>