



tic

Cuadernos de desarrollo aplicados a las TIC

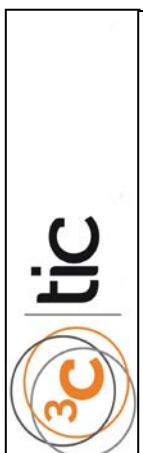


Edición nº 9
Volumen 3 Número 2
Junio - Septiembre 2014
ISSN: 2254 - 6529
Publicación trimestral

INDEXACIÓN	CATÁLOGOS
  CRUE REBIUN Red de Bibliotecas Universitarias  DIRECTORY OF OPEN ACCESS JOURNALS  	 Universitat d'Alacant Universidad de Alicante  UNIVERSIDAD POLITECNICA DE VALENCIA  UNIVERSIDAD PONTIFICIA ICAI ICADE COMILLAS M A D R I D  BASE DE DATOS ISOC   Biblioteca Valenciana  BIBLIOTECA NACIONAL BN  CONSEJO SUPERIOR DE INVESTIGACIONES CIENTÍFICAS 

PÚBLICO AL QUE VA DIRIGIDA LA REVISTA:

- **Personal investigador.**
- **Doctorandos.**
- **Profesores** de universidad.
- **Oficinas de transferencia de resultados de investigación. (OTRI)**
- **Empresas** que desarrollan **labor investigadora** y quieran publicar alguno de sus estudios.

 <p>tic 3C</p>	<p><i>3c Tic, cuadernos de desarrollo aplicados a las TIC</i> <i>Periodicidad trimestral</i> <i>Edición nº 9</i> <i>Volumen 3 Número 2 (Junio – septiembre 2014)</i> <i>Tirada nacional e internacional</i> <i>Artículos revisados por el método de evaluación por pares de doble ciego.</i> <i>ISSN: 2254 – 6529</i> <i>Depósito legal: A 298 - 2012</i></p>	<p>Editorial: Área de Innovación y Desarrollo, S.L. Empresa de transferencia del conocimiento al sector empresarial. Alcoy, Alicante (España) C/ Santa Rosa 15, nº 3 Tel: 965030572 E-mail editor: info@3ciencias.com</p>
---	---	--

NORMATIVA DE PUBLICACIÓN

- Los artículos, que **serán inéditos**, (no podrán haberse publicado anteriormente) tendrán una extensión máxima de 3.500 palabras, incluyendo notas a pie de página y bibliografía, aunque se apreciarán extensiones más breves. No deberá utilizarse un número excesivo de referencias bibliográficas. El resumen no excederá de 200 palabras.
- El título del artículo deberá estar expresado tanto en castellano como en inglés.
- Los artículos deberán estar escritos en castellano.
- Cada artículo deberá ir precedido de un pequeño resumen, en castellano e inglés, y de cinco palabras clave en ambos idiomas. Además se incorporará la clasificación del trabajo conforme a los descriptores utilizados por el Journal Economic Literature.
- Se valorará la inclusión de cuadros y gráficos que apoyen las tesis desarrolladas en el artículo.
- Deberá aparecer el nombre del autor/es en la primera hoja, junto a su titulación académica oficial y la universidad, institución o empresa en la que presten sus servicios.
- Las referencias irán al final del artículo bajo el epígrafe Referencias bibliográficas, ordenadas alfabéticamente por autores y de acuerdo con el siguiente orden: nombre (en minúsculas) del autor o autores, iniciales de los apellidos, año de publicación (entre paréntesis y distinguiendo a, b, c, en caso de que el mismo autor tenga más de una obra citada en el mismo año), título del artículo (entre comillas) y título de la revista a la que pertenece el artículo (en cursiva o subrayado).
- No se admitirán artículos con errores ortográficos. Los contenidos de los artículos deben ser cuidadosamente leídos y revisados antes de su envío, tanto por el autor como por un amigo o colega crítico.
- Los originales estarán editados electrónicamente en formato "Word" o compatible y a color.
- Las imágenes de la publicación se enviarán en formato jpg.
- La revista se reserva la posibilidad de editar y corregir los artículos, incluso de separar y recuadrar determinadas porciones del texto particularmente relevantes o llamativas, respetando siempre el espíritu del original.
- Se debe evitar utilizar un lenguaje de corte excesivamente especializado, en beneficio de una más fácil comprensión de las ideas expuestas y en la medida de lo posible, el abuso en la utilización de lenguaje y funciones matemáticas.
- Los autores deberán ceder los derechos de publicación de los artículos a ÁREA DE INNOVACIÓN Y DESARROLLO, S.L.

RULES OF PUBLICATION / INSTRUCTIONS TO AUTHORS

- The articles, which are unpublished, have a maximum length of 3,500 words, including footnotes and bibliography page, even shorter extensions appreciate. You should not use too many references. The abstract should not exceed 200 words.
- The title of the article should be expressed both in Castilian and English.
- The articles should be written in Spanish.
- Each article should be preceded by a short summary, in Spanish and English, and five key words in both languages. Furthermore incorporate job classification according to the descriptors used by the Journal of Economic Literature.
- It will assess the inclusion of charts and graphs that support the thesis developed in the article.
- You should see the name of the author/s on the first page, along with their academic qualifications and university official, institution or company in which they are employed.
- References appear at the end of the article under the heading References , arranged alphabetically by authors and according to the following order : name (lowercase) of author , initials of the last names , year of publication (in brackets and distinguishing , b , c , in the event that the author has more than one work cited in the same year) , title of article (in quotation marks) and title of the journal to which the article belongs (in italics or underlined) .
- May not be misspelled items. The contents of the articles should be carefully read and reviewed prior to shipment, both by the author as a critical friend or colleague.
- The originals will be published electronically in "Word" or compatible and color.
- The images of the publication will be sent in jpg format.
- The magazine reserves the right to edit and correct items, including certain portions separate and square up the particularly relevant or bold text, respecting the spirit of the original.
- Avoid using excessively cutting a language specialist, the benefit of an easier understanding of the ideas and to the extent possible, the use abuse language and mathematical functions.
- The authors must assign the rights to the articles published INNOVATION AND DEVELOPMENT AREA, SL

SUMARIO

ARTÍCULOS:

PASEO VIRTUAL DEL MUSEO PROVINCIAL DE GRANMA.....	66
VIRTUAL TOUR OF THE PROVINCIAL MUSEUM OF GRANMA.....	66
Saylin Pompa Núñez y Dainel Arzuaga Rodríguez	
 SISTEMA DE GESTIÓN DE LA INFORMACIÓN PARA EL DESARROLLO LOCAL	77
INFORMATION MANAGEMENT SYSTEM FOR LOCAL DEVELOPMENT.....	77
Juliett Martínez López, Raúl Rubén Fernández Aedo y Eglys Pérez Ugartemendía	
 PROPIUESTA DE PROCEDIMIENTO PARA REALIZAR PRUEBAS DE CAJA BLANCA A LAS	
APLICACIONES QUE SE DESARROLLAN EN LENGUAJE PYTHON.....	89
PROPOSED PROCEDURE FOR WHITE BOX TESTING APPLICATIONS DEVELOPED IN	
PYTHON LANGUAGE	89
Eduardo Salazar Martínez	
 APLICACIONES INFORMÁTICAS PARA LA GESTIÓN ECONÓMICA FINANCIERA Y EL	
ASEGURAMIENTO DE LA UNIVERSIDAD DE LA HABANA.....	115
COMPUTER APPLICATIONS FOR FINANCIAL MANAGEMENT AND FINANCIAL	
ASSURANCE OF THE UNIVERSITY OF LA HABANA.....	115
Viviana Duro Novoa	



Recepción: 04 de noviembre de 2013
Aceptación: 10 de marzo de 2014
Publicación: 26 de junio de 2014

PASEO VIRTUAL DEL MUSEO PROVINCIAL DE GRANMA

VIRTUAL TOUR OF THE PROVINCIAL MUSEUM OF GRANMA

Saylin Pompa Núñez ¹

Dainel Arzuaga Rodríguez ²

1. Ingeniera en Ciencias Informáticas. Facultad Regional Granma de la Universidad de las Ciencias Informáticas, Especialista de Realidad Virtual del departamento Web y Multimedia. Cuba. E-mail: spnunez@grm.uci.cu
2. Máster en Ciencias. Licenciado en Matemática. Profesor del Departamento de Ciencias Básicas. Facultad de Ciencias Informáticas de la Universidad de Granma. Cuba. E-mail: dainel@grm.uci.cu

RESUMEN

Una de las formas que permite que personas de todo el mundo conozcan sitios importantes es la realización de visitas virtuales sobre la web, haciendo uso de la realidad virtual, tecnología de interfaz avanzada entre varios usuarios y los sistemas de cómputos.

La presente investigación tiene como base diseñar e implementar un paseo virtual por diferentes salas del Museo Provincial de Granma, el mismo está comprendido por recorridos que muestren el rico contenido patrimonial de dicho museo.

ABSTRACT

One way that allows people around the world to know important sites, is the realization of virtual tours on the web, using virtual reality, advanced interface technology's between multiple users and computing systems.

This research is based on designing and implementing a virtual walk through different rooms of the Provincial Museum of Granma, the same is comprised of tours that showcase the rich heritage of the museum content.

PALABRAS CLAVE

Paseo virtual, realidad virtual, visitas virtuales.

KEYWORDS

Virtual walk, virtual reality, virtual visits.

INTRODUCCIÓN

En la actualidad, la difusión y protección del Patrimonio constituye uno de los principales retos para identificar las raíces históricas de la nación, convirtiéndose estas acciones en procesos claves para la construcción de nuestra cultura; en la medida en que su realización nos posibilita verificar actitudes, desarrollar comportamientos y cimentar valores adquiridos en la producción cultural a través de los años.

El patrimonio a su vez, es apropiable como objeto de disfrute, motiva al aprendizaje de la historia, estimula el desarrollo intelectual, crea sentimientos patrióticos y satisface las necesidades espirituales, propiciando un ambiente creativo y agradable. Por tanto, los bienes patrimoniales, constituyen una herramienta especial para que todos conozcamos nuestro pasado, comprendamos el presente y nos proyectemos hacia el futuro. A partir de esto, los organismos encargados de administrar y gestionar el patrimonio, utilizan como su principal centro de conservación y difusión a los museos.

Actualmente, los museos generan varios procesos, tanto administrativos y organizativos, como de corte científico-metodológicos. Siendo estos últimos los encargados de reunir las colecciones, estudiarlas, documentarlas, conservarlas, interpretarlas y exhibirlas en función de la temática del museo; además de constituir los procesos que mantienen la perdurabilidad de las instituciones museísticas en la sociedad.

Con el desarrollo de las nuevas tecnologías, las personas han encontrado varias formas de crear y modelar un mundo virtual que se asemeje a la realidad, con el fin de experimentar mecanismos, experiencias y sensaciones para evitar errores que conlleven a gastos incalculables de dinero y pérdidas de vidas humanas. La alternativa mundialmente para lograr lo antes mencionado, se conoce como realidad virtual. Consiste en un sistema tecnológico, basado en el empleo de ordenadores de alto rendimiento y dispositivos especialmente diseñados, cuya meta es recrear virtualmente la realidad.

La realidad virtual se ha extendido a muchas áreas de interés social, como la salud, la educación, la industria, los videojuegos, la astronomía, el sector militar y la cultura. En esta última área se ha visto con mayor fuerza la difusión del patrimonio cultural a través de paseos virtuales, los cuales han encontrado una manera más intuitiva de visualizar sitios de interés social, logrando que estén al alcance de todos, dejando atrás las restricciones que ocasiona el traslado hacia la ubicación física del sitio en cuestión.

El Museo Provincial de Granma es una institución que atesora una parte del patrimonio cultural de la provincia, actualmente es centro de visitas de turistas que recorren la hermosa ciudad de Bayamo. Con el avance de nuevas tecnologías son muchas las personas que buscan información del destino a visitar en la web, sin embargo es insuficiente la información del patrimonio que se atesora en dicho museo.

El **objetivo** de la investigación es desarrollar un paseo virtual con imágenes panorámicas del Museo Provincial de Granma, que permita divulgar el patrimonio de esta institución a través de la web.

DESARROLLO

FLUJO ACTUAL DEL PROCESO

La Facultad Regional de Granma (FRG) cuenta con un Centro de Desarrollo (CD) cuyo objetivo es crear soluciones informáticas para diferentes sectores como son: salud, educación, cultura, etc. Dicho centro está organizado en dos líneas temáticas de trabajo y varios grupos de desarrollo. Además del sector de cultura, Patrimonio Cultural es uno de los principales clientes y evidencia de esto es la existencia de un acuerdo de colaboración mediante el cual, la FRG desarrolla aplicaciones informáticas que contribuyen a la promoción, divulgación y valorización del patrimonio de la provincia. En la actualidad no existe una manera viable donde se muestre y se divulgue el patrimonio cultural que se atesora en el Museo Provincial de Granma, ya que el mismo no explota las potencialidades de las Tecnologías de las Información y las Comunicaciones (TIC) para la difusión de su patrimonio. Para conocer el museo y sus colecciones, obligatoriamente, hay que dirigirse a la institución y resulta difícil incluso para las mismas personas de la provincia de Granma, por ello se hace necesario el desarrollo inmediato de una aplicación que solucione este problema. A continuación se detalla la metodología, herramientas y lenguajes para la construcción del software.

METODOLOGÍA DE DESARROLLO DE SOFTWARE

En la elaboración de un producto informático se utilizan las metodologías de desarrollo de software, encaminadas para ayudar a realizar el producto con la calidad requerida y que cumpla con las exigencias del cliente. Las metodologías imponen un proceso disciplinado sobre el desarrollo de aplicaciones con el principal objetivo de hacerlo más predecible y eficiente.

“Una de las metodologías ágiles es la SXP, que es un híbrido cubano creado de la unión entre las metodologías SCRUM y eXtreme Programming (XP). Esta metodología toma de la XP la parte ingenieril y la gestión de proyectos de la SCRUM. Ofrece además una estrategia tecnológica, a partir de la introducción de procedimientos ágiles que permiten actualizar los procesos de software para el mejoramiento de la actividad productiva, fomentando el desarrollo de la creatividad, aumentando el nivel de preocupación y responsabilidad de los miembros del equipo y ayudando al líder del proyecto a tener un mejor control del mismo. Consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto. ” [1]

Consta de cuatro fases llamadas **Planificación - Definición, Desarrollo, Entrega y Mantenimiento**. De cada una de ellas se despliegan siete flujos de trabajo (Concepción inicial, Captura de requisitos, Diseño, Implementación, Prueba, Entrega de la documentación, Soporte e Investigación).

Se escogió la metodología SXP porque está especialmente indicada para proyectos de pequeños equipos de trabajo, rápido cambio de requisitos o requisitos imprecisos, donde existe un alto riesgo técnico. Se orienta a una entrega rápida de resultados y una alta

flexibilidad.

LENGUAJES

Lenguaje de Marcas de Hipertexto (HTML) es un lenguaje de composición de documentos y especificación de ligas de hipertexto que define la sintaxis y coloca instrucciones especiales que no muestra el navegador, aunque sí le indica cómo desplegar el contenido del documento, incluyendo texto, imágenes y otros medios soportados, las cuales conectan diferentes documentos, ya sea en su computadora o en otras, así como otros recursos de Internet.

HTML se utiliza por presentar cualidades que son favorables en el desarrollo de la arquitectura, entre las que se destacan:

- Etiquetas que posibilitan señalar la semántica del contenido de un documento.
- Los documentos, que pueden incluir texto, imagen, sonido y video.

Hojas de Estilo en Cascada (CSS), es un lenguaje de hojas de estilos creado para controlar la presentación de los documentos electrónicos definidos con HTML. CSS es la mejor forma de separar los contenidos y su presentación, y es imprescindible para la creación de páginas web complejas.

El empleo de CSS facilita al implementador tener el dominio sobre el estilo y formato de los documentos, su tamaño, tipo de letra, color, separación entre titulares y párrafos, etc. Permite dar estilo a múltiples páginas web al mismo tiempo para facilitar la reutilización de código y optimización del sistema.

JavaScript es un lenguaje de programación interpretado, es utilizado principalmente en páginas web dinámicas. Los programas hechos en JavaScript se pueden probar en cualquier navegador sin necesidad de procesos intermedios. Una de las ventajas es que todos los navegadores modernos interpretan el código JavaScript integrado dentro de dichas páginas.

HERRAMIENTAS

Geany es un editor de texto con características básicas de un entorno de desarrollo integrado (IDE). Es capaz de reconocer archivos HTML, Python, JavaScript, entre muchos otros. Aporta varias ventajas, como el resaltado de sintaxis o poder usar la función de autocompletado. De igual manera, incluye las herramientas necesarias para compilar y ejecutar los archivos que son trabajados mediante este pequeño editor de texto. Destacar además que es válida su selección gracias a la posibilidad de poder explotar características propias de este entorno, tales como: soporte multidocumento, coloreado de sintaxis y facilidad de uso.

GNU Image Manipulation Program (GIMP) es una herramienta para el tratamiento de edición de imágenes digitales en forma de mapa de bits, tanto dibujos como fotografías. Se seleccionó esta herramienta por ser un programa libre y gratuito, además por el amplio conocimiento y fácil manejo que tiene el equipo de desarrollo con dicha herramienta.

Hugin es una herramienta para la creación de imágenes panorámicas a partir de fotos individuales. No es cuestión simplemente de cargar las imágenes y dejar al programa hacer su trabajo, sino que a través de una gran cantidad de opciones permite controlar hasta el mínimo ajuste del resultado final, de tal manera que se obtienen panorámicas con buena calidad.

Los visores de imágenes panorámicas tienen la misión de mostrar un panorama que se distingue de una imagen convencional por el amplio horizonte visual que cubre. Estos se han expandido hasta permitir la existencia de mundos virtuales en los que se puede ingresar y desplazarse para observar diferentes puntos de vista de una escena.

En la presente investigación el visor de imágenes panorámicas que se utilizó fue PIViewer, ya que solamente utiliza estándares de desarrollo web. Esto garantiza que no sea necesario instalar programas adicionales en el navegador para visualizar el paseo. El mismo ha sido creado en la FRG, donde se le agregaron funcionalidades para mostrar al cliente un producto de alta calidad.

ESPECIFICACIÓN DE REQUISITOS DEL SISTEMA

La ingeniería de requerimientos es uno de los procesos más importantes en el desarrollo de software sin tener en cuenta la metodología que se utilice. Su objetivo es identificar y documentar los requerimientos del sistema, para tener una visión más clara de lo que se debe hacer. Los requisitos son las condiciones y capacidades que tienen que ser alcanzadas por un sistema que necesita un usuario para resolver un problema o lograr un objetivo. Los mismos deben ser especificados por escrito como una característica del sistema a entregar y lo más conciso posible. A continuación se detallan las funcionalidades del producto:

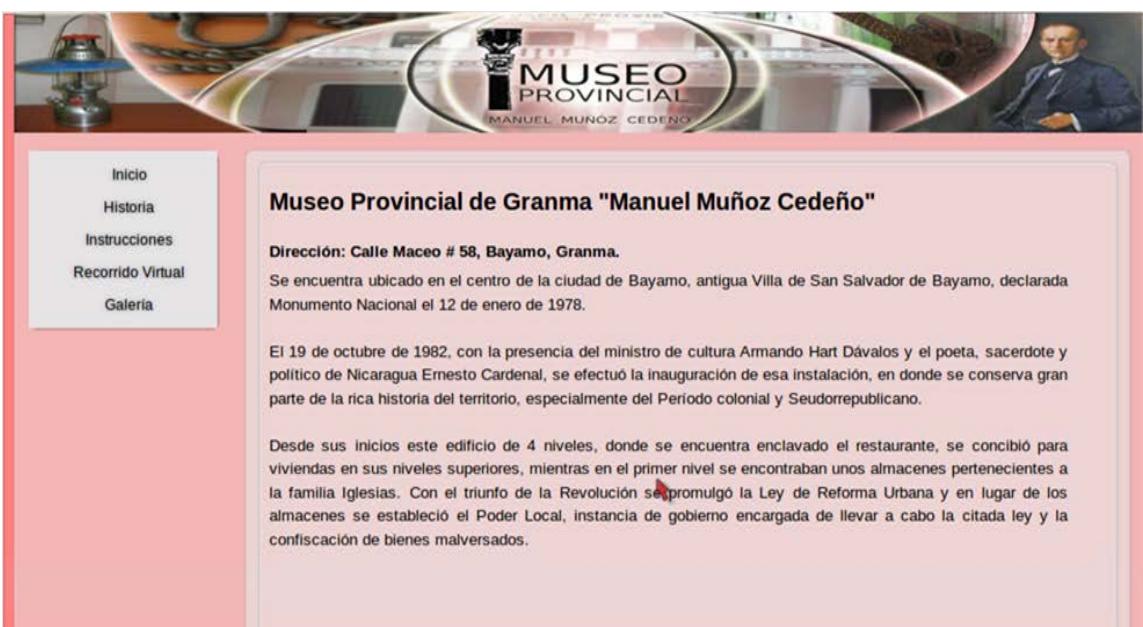


Gráfico 1: Mostrar Información del Museo



Gráfico 2: Acceder a las distintas salas del museo

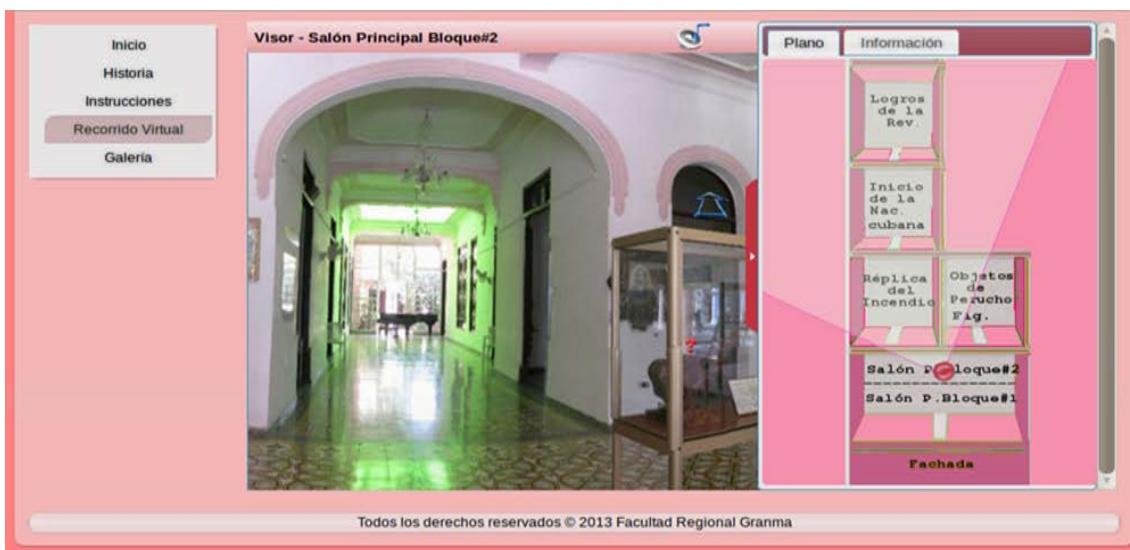


Gráfico 3: Mostrar ubicación y campo de visión del usuario

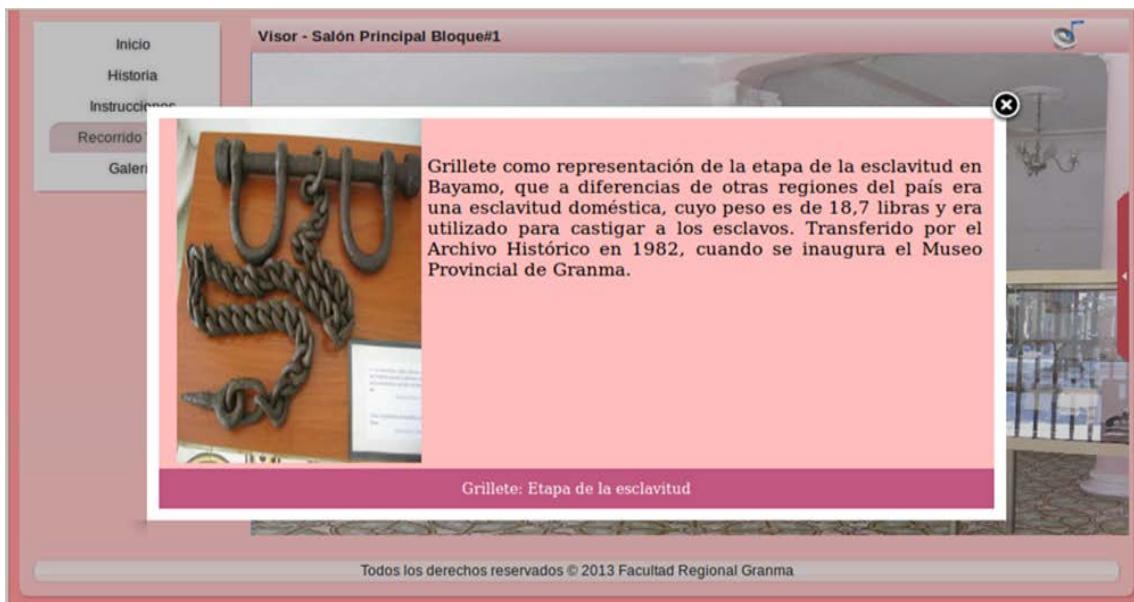


Gráfico 4: Mostrar los objetos más significativos de cada sala



Gráfico 5: Mostrar descripción de las salas visitadas

RESULTADOS Y DISCUSIÓN

ASEGURAMIENTO DE LA CALIDAD DEL SOFTWARE

Una de las últimas fases del ciclo de vida de un software antes de entregarlo para su explotación, es la fase de pruebas. El objetivo específico es encontrar la mayor cantidad de errores posibles, para proceder a corregirlos. Durante esta fase, el sistema o componente es ejecutado bajo condiciones o requerimientos especificados, los resultados son observados y registrados para realizar una evaluación.

Una estrategia de prueba vela que el producto que se está construyendo reúna los requerimientos de la lógica del negocio que el cliente ha pedido mediante el contrato de desarrollo del software. Es la línea de guía del equipo de pruebas de un proyecto. Según Pressman. “Una estrategia de prueba del software integra los métodos de diseño de caso de pruebas del software en una serie bien planeada de pasos que desembocará en la eficaz construcción de software.” [2]

ESTRATEGIA DE PRUEBA

Las estrategias de pruebas permiten identificar las características de calidad que deben ser evaluadas en un software. Son creadas para reunir las ideas más representativas del proceso de pruebas que se llevará a cabo. Estas proporcionan una guía para lograr que las pruebas se realicen con la calidad requerida y que arrojen los resultados esperados.

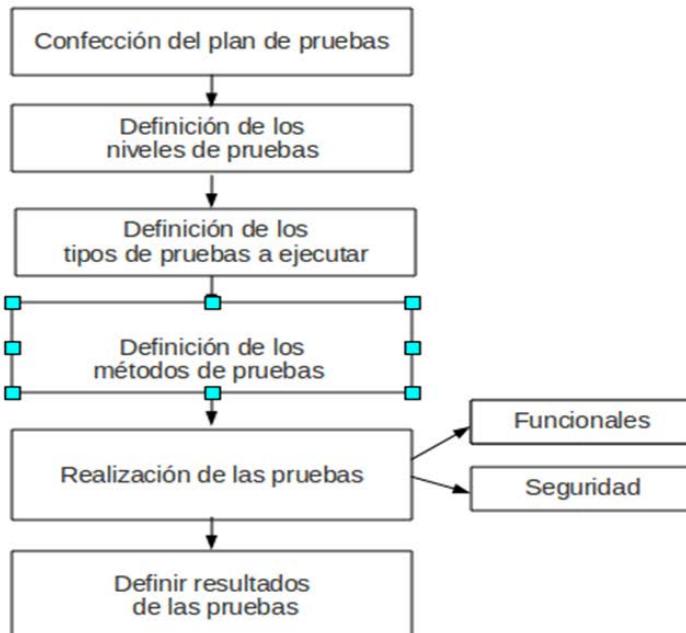


Gráfico 6: Estrategia de prueba. Fuente: Elaboración propia.

CONCLUSIONES

Se demostró la importancia de desarrollar una aplicación web en entornos libres que responda a la necesidad que la originó, siendo un sistema capaz de brindar información relevante del Museo Provincial de Granma.

La aplicación desarrollada trajo consigo los beneficios que se exponen a continuación:

- El sitio web contribuyó a conservar y divulgar el patrimonio tangible e intangible que atesora la institución y conocer sobre la historia del Museo Provincial de Granma.
- Permitió reforzar el trabajo de animación sociocultural y la divulgación del patrimonio del museo.
- Contribuyó al impacto de la FRG en la informatización de la región oriental del país.
- Permitió enriquecer la sección del portal provincial de patrimonio dedicada a los museos.

REFERENCIAS BIBLIOGRAFICAS

- [1] Penalver, G., Meneses, A. y García, S. *SXP, Metodología Ágil para el Desarrollo de Software*. Habana: s.n., 2010.
- [2] Pressman, Roger. "INGENIERÍA DEL SOFTWARE: Un enfoque práctico". 6a. ed.2005. Capítulo 13. *Estrategias de pruebas del software*. ISBN 9701054733.

Recepción: 03 de diciembre de 2013

Aceptación: 14 de abril de 2014

Publicación: 26 de junio de 2014

SISTEMA DE GESTIÓN DE LA INFORMACIÓN PARA EL DESARROLLO LOCAL

**INFORMATION MANAGEMENT SYSTEM FOR LOCAL
DEVELOPMENT**

MsC. Juliett Martínez López¹

Dr. C. Raúl Rubén Fernández Aedo²

Dra. C. Eglys Pérez Ugartemendía³

1. Profesora asistente. Facultad de Ciencias Informáticas. Universidad Máximo Gómez Báez de Ciego de Ávila. Cuba.
2. Profesor titular. Centro de Estudio de Gestión de la Información y el Conocimiento. Universidad Máximo Gómez Báez de Ciego de Ávila. Cuba. E-mail: aedo@informatica.unica.cu
3. Profesora auxiliar. Facultad de Ingeniería. Universidad Máximo Gómez Báez de Ciego de Ávila. Cuba.

RESUMEN

La gestión de la información es un elemento esencial para la toma de decisiones y el desarrollo local en la provincia Ciego de Ávila, sin embargo a partir de un estudio realizado se pudo analizar que el manejo de la información por parte de individuos, empresas y organizaciones presentaba deficiencias y debilidades. Las tecnologías de la información han puesto de manifiesto la complejidad social de las mismas y su dinamismo en todos los niveles de la sociedad. El trabajo que se propone se auxilia de estas tecnologías para el desarrollo de un sistema informático que contribuya a disminuir las insuficiencias detectadas. La corroboración de los resultados muestra la validez y utilidad de la herramienta propuesta para llevar un control estricto que favorezca la toma de decisiones y con ello el desarrollo local en general en la provincia Ciego de Ávila.

ABSTRACT

Information management is an essential element for decision making and development in the province of Ciego de Ávila. However, from a study it was possible to analyze that the management of information by individuals, companies and organizations had deficiencies and weaknesses. Information technologies have revealed their social complexity and dynamism in all levels of society. The proposed work is aided by these technologies for the development of a computer system to help reduce the weaknesses observed. Corroboration of the results shows the validity and usefulness of the tool proposed to maintain tight control, enhancing the decision-making and thus overall local development in the province of Ciego de Ávila.

PALABRAS CLAVE

Sistema, gestión, información, desarrollo, local.

KEYWORDS

System, management, information, development, local.

INTRODUCCIÓN

El desarrollo local es un proceso mediante el cual los gobiernos locales establecen iniciativas y promueven actividades económicas eficientes y eficaces de forma coordinada con todos los agentes políticos y sociales en proyectos conjuntos que influyen decisivamente en el sector productivo, incentivándolos con el objetivo principal de rediseñar la estructura socio-económica del territorio y así incrementar los valores productivos, la eficiencia en la gestión y la efectividad social. Como elemento clave para el logro del desarrollo local debe existir un adecuado uso de la información por parte de individuos, empresas y organizaciones. La información, está constituida por un grupo de datos ya supervisados y ordenados, para construir un mensaje basado en un cierto fenómeno o ente que permita resolver problemas y tomar decisiones, ya que su aprovechamiento racional es la base del conocimiento.

Sin embargo en la provincia Ciego de Ávila existen deficiencias en la selección de alternativas de solución de los problemas territoriales, por parte del gobierno e instituciones, en ocasiones basada en la experiencia individual o en el momento que aparece la necesidad. Además en ocasiones se detecta demora en la respuesta, por parte del gobierno o instituciones, para resolver los problemas presentados en las localidades. Todo esto afecta la toma de decisiones para el desarrollo local en la provincia.

Varios autores han investigado en esta área como [1], [2], [3], [4] y [5]. En Cuba se han realizado estudios en esta línea, incluso por profesores de la Universidad de Ciego de Ávila Máximo Gómez Báez, como el Dr. Elme Carballo Ramos y el Dr. Raúl Fernández Aedo. Sin embargo aún se presentan limitaciones en la toma de decisiones para el desarrollo local en la provincia Ciego de Ávila. Por lo que este trabajo tiene como objetivo desarrollar un sistema de gestión de información desde una dinámica de relaciones de coordinación e integración, con el uso de lenguajes de alto nivel y bases de datos, para mejorar la toma de decisiones y contribuir al desarrollo local.

DESARROLLO

La información permite resolver problemas y tomar decisiones, ya que su uso racional es la base del conocimiento. Es un fenómeno que proporciona significado o sentido a las cosas y aumenta el conocimiento sobre algo, un mensaje con significado en un determinado contexto, disponible para uso inmediato y que proporciona orientación a las acciones por el hecho de reducir el margen de incertidumbre con respecto a nuestras decisiones [6].

En las organizaciones y empresas la información ha adquirido desde hace varios años un papel esencial. Se le considera como un activo fundamental, de tal manera que sin su presencia no se entiende una eficaz toma de decisiones. El personal de cualquier organización o empresa, orientado a mejorar los objetivos y propósitos para el desarrollo local, necesita información externa e interna para tomar decisiones y para poder planificar en el marco de sus estrategias y necesidades. Entiéndase por toma de decisiones, como la selección de un curso de acciones entre varias alternativas [7].

El desarrollo territorial constituye un proceso localizado de cambio socio-económico sostenido que tiene como finalidad última, el progreso permanente del territorio, la localidad, la comunidad y los individuos que la habitan. En este proceso, la información se ha convertido en un recurso indispensable en el funcionamiento de las empresas y organizaciones para la toma de decisiones.

Varios autores han definido la gestión de la información, como el proceso mediante el cual se obtienen, despliegan o utilizan recursos básicos (económicos, físicos, humanos y/o materiales) para manejar información dentro de las empresas u organizaciones y para la sociedad a la que sirve [4]. Una adecuada gestión de la información, posibilita reducir los riesgos en la administración del territorio, como son la toma de decisiones apresuradas, tardías o inconsistentes, que ocasionan pérdidas y reducen su competitividad o desarrollo. Obtener la información necesaria, con la calidad requerida, es una premisa indispensable para la supervivencia de la sociedad, si se considera que las empresas u organizaciones acortan cada vez más sus ciclos estratégicos y que la toma de decisiones, así como el cambio es continua.

Los sistemas de gestión de información o sistemas de información [8], transforman datos de entrada, los procesa, los almacena para su posterior uso por los usuarios internos y externos que estén interesados. La información es el resultado del procesamiento de los datos. Los sistemas de información tienen como objetivo satisfacer las necesidades de información de sus usuarios mediante la integración de los flujos de información existentes en las empresas u organizaciones. Junto a las tecnologías de la información y las redes de información compartida, generan una revolución que favorece el acceso rápido a la información interna y externa, formal e informal, previene las duplicaciones de información, produce el máximo rendimiento en el uso de la información existente y evitan la territorialidad de información.

Se denominan Tecnologías de la Información y las Comunicaciones, en adelante TIC [9], al conjunto de tecnologías que permiten la adquisición, producción, almacenamiento, tratamiento, comunicación, registro y presentación de informaciones, en forma de voz,

imágenes y datos contenidos en señales de naturaleza acústica, óptica o electromagnética. Las TIC incluyen la electrónica como tecnología base que soporta el desarrollo de las telecomunicaciones, la informática y el audiovisual. Constituyen el núcleo central de una transformación multidimensional que experimenta la economía y la sociedad. Ellas ejercen una notable influencia en los procesos de generación, producción, transmisión, difusión y organización de la información y el conocimiento, lo que exige de una innovación tecnológica y organizacional permanente, con vistas a garantizar niveles altos de competitividad y de respuesta a una demanda mayor de las localidades.

Ante el desarrollo local, la gestión de la información se convierte en un aspecto estratégico para las empresas u organizaciones que se insertan en el actual entorno y asumen las nuevas tecnologías de información y comunicación. Las que recurren a la implementación de coherentes sistemas de gestión de información, espacios e infraestructuras para disponer de su propia información, compartir sus recursos y poseer canales de comunicación rápidos y eficientes, que colaboren con el desarrollo del trabajo y la toma de decisiones.

SITUACIÓN ACTUAL DEL PROCESO DE GESTIÓN DE LA INFORMACIÓN PARA EL DESARROLLO LOCAL EN LA PROVINCIA CIEGO DE ÁVILA

Tras un estudio realizado, en función del desarrollo local, en conjunto con los directivos de los consejos de administración y los funcionarios del poder popular de los municipios en la provincia Ciego de Ávila, se pudo constatar entre otras cosas que:

- ✓ La información manejada diariamente no es suficiente para poder tomar decisiones certeras, oportunas, rápidas y eficientes en cada caso, lo que incide de manera negativa en el cumplimiento de tareas y en el desarrollo local de sus territorios.
- ✓ Se necesita información para el control interno, temas financieros, jurídicos y económicos actualizados que permita valorar desde una mejor óptica el trabajo de los directivos.
- ✓ No se dispone de modelos oficiales que permitan enviar la información solicitada por los organismos superiores.
- ✓ No existe una herramienta informática para viabilizar el proceso de gestión de la información, por lo que esta se lleva en algunos casos en documentos Word u hojas de Excel y en otros casos, la mayoría de las ocasiones, se realiza de forma manual en papeles que luego se archivan, lo que trae consigo errores de transcripción frecuentes, demora en la información que se necesita y en algunos casos desconocimientos. Los documentos archivados se pueden prestar a la pérdida o el deterioro, además cuando se quiere conocer algún dato o información es necesario emplear gran cantidad de tiempo para buscar en los documentos, al tener que consultar información dispersa.
- ✓ La información recibida de sus localidades, en la mayoría de los casos, es insuficiente e incompleta, lo que lleva a tomar decisiones por valoraciones subjetivas.

Todo lo anterior refleja que la información manejada es incompleta, presenta errores en muchos casos y es insuficiente para lograr una toma de decisiones correcta por parte de los directivos además existe poca actualización en temas primordiales para el desarrollo local. Esto revela la necesidad de un sistema informático de gestión de la información para el desarrollo local que contribuya, desde una dinámica de relaciones de coordinación e integración, contar con una infraestructura para disponer de su propia información, compartir sus recursos y poseer canales de comunicación rápidos y eficientes, que colaboren con el desarrollo del trabajo y la toma de decisiones de la provincia de Ciego de Ávila.

DESARROLLO DEL SISTEMA DE GESTIÓN DE LA INFORMACIÓN PARA EL DESARROLLO LOCAL (SGIDL)

Entre los medios y herramientas más importantes, usados para un buen procesamiento de la información, se encuentran las llamadas Aplicaciones Web y las Aplicaciones de Escritorio, ambas favorecen a los usuarios, desde su descubrimiento, hasta convertirse actualmente en sistemas que, adaptados a características concretas y particulares, han pasado a formar parte elemental y punto de atención de cualquier organización. El desarrollo de una aplicación Web es la elección adecuada para lograr los objetivos propuestos, ya que estas emplean la arquitectura cliente-servidor, muy ventajosa cuando se manejan datos que se comparten entre usuarios de diferentes localidades, organizaciones y empresas.

Las metodologías de diseño y de desarrollo de aplicaciones basadas en UML, son conocidas y utilizadas muy a menudo. Sin embargo, estas metodologías y herramientas no proveen soporte específico para modelar las características propias de las aplicaciones Web. Las metodologías Web surgen con el propósito de ser útiles para afrontar los retos específicos que presentan las aplicaciones Web. OO-H (del inglés *Object-Oriented Hipertext*) [10] define un enfoque orientado a objetos para el desarrollo de aplicaciones Web. Con respecto al camino utilizado para la construcción del hipertexto, utiliza un enfoque orientado a procesos, dado que define guías para derivar el hipertexto de la aplicación (páginas Web y enlaces entre ellas) a partir del modelo flujo de proceso. Utiliza UML de manera parcial, pues propone utilizar algunos diagramas propios, como el diagrama de acceso navegacional.

Para el diseño del SGIDL, se utilizó la metodología OOH. Con su uso queda establecida la arquitectura para el desarrollo del software, obteniéndose como principales artefactos los modelos de Requisitos, de Dominio, de Análisis de Navegación y el Diseño de Presentación.

HERRAMIENTAS INFORMÁTICAS UTILIZADAS EN EL DESARROLLO DEL SGIDL.

A continuación se presentan los principales lenguajes, herramientas y técnicas empleadas en el desarrollo de la aplicación Web, siguiendo la filosofía de uso de software libre.

- ✓ **PHP (Hypertext Pre-processor):** Es un lenguaje de programación potente del lado del servidor especialmente diseñado para la programación Web dinámica. Tiene soporte con un conjunto de motores de base de datos. Puede ser desplegado en la mayoría de los servidores Web y en casi todos los Sistemas Operativos y plataformas sin costo alguno [11].
- ✓ **JavaScript:** es un lenguaje de programación interpretado [12], por lo que no es necesario compilar los programas. Es un lenguaje del lado del cliente.
- ✓ **UML (Lenguaje Unificado de Modelado):** Es un lenguaje de propósito general para el modelado Orientado a Objetos. Se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software; para entender, diseñar, configurar, mantener y controlar la información sobre los sistemas a construir [13].
- ✓ **CSS (Cascading Style Sheets):** Es un lenguaje de estilo usado para definir la presentación de un documento estructurado escrito en HTML (XHTML) [14]. Su objetivo esencial es separar la estructura de un documento de su presentación.
- ✓ **HTML (HyperText Markup Language):** Es el lenguaje en que se escriben los documentos que hoy existen en el *World Wide Web* [15]. Cuando se accede a uno de estos documentos, el navegador lo interpreta y lo muestra.
- ✓ **PostgreSQL:** Es un potente motor de bases de datos, que tiene prestaciones y funcionalidades equivalentes a muchos gestores de bases de datos comerciales [16]. Es un sistema de gestión de base de datos relacional orientada a objetos de software libre. Soporta integridad referencial, la cual es utilizada para garantizar la validez de los datos de la base de datos. Usa una arquitectura proceso-por-usuario cliente/servidor.
- ✓ **Adobe Dreamweaver:** Es una aplicación destinada a la construcción, diseño y edición de sitios y aplicaciones Web basados en estándares [17].

En la figura 1 y la figura 2 se muestran algunas páginas del SGIDL relacionadas con la gestión de parámetros medibles por tipo de empresa y de los datos de cada entidad en particular, estas recogen algunas de las principales funcionalidades del sistema.



Fig. 1. Gestionar etiquetas por tipo de entidad

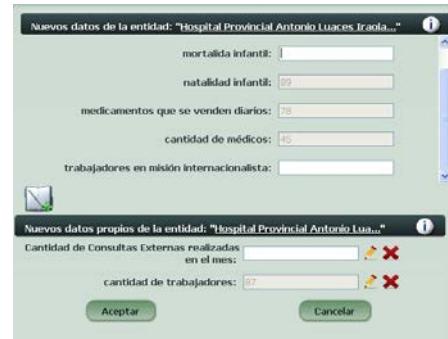


Fig. 2. Gestionar datos por entidad

CORROBORACIÓN DEL SISTEMA DE GESTIÓN DE LA INFORMACIÓN PARA EL DESARROLLO LOCAL

Siguiendo la metodología utilizada, se realizan pruebas al sistema que permiten limar los posibles errores que emergen durante el desarrollo de software y comprobar que el mismo funciona correctamente. Los principios de interfaz, el tratamiento de errores, los medios de ayuda y la seguridad de la que se provee el sistema, constituyen garantías al cliente de la confiabilidad de la herramienta desarrollada.

Para la corroboración del sistema se utiliza el criterio de expertos [18] y como parámetros medibles los siguientes aspectos: fundamentación, intencionalidad, definición de los objetivos, aplicabilidad, necesidades del tema, vinculación con otras actividades y evaluación de la propuesta. Los indicadores propuestos se someten al criterio de los expertos seleccionados y finalmente ellos coinciden en que el software propuesto, reúne todas las condiciones necesarias y pertinentes para llevar un control estricto que favorezca la toma de decisiones y con ello el desarrollo local en general en la provincia Ciego de Ávila.

CONCLUSIONES

- ✓ Existen limitaciones en el proceso de gestión de la información del desarrollo local en la provincia Ciego de Ávila. Los estudios realizados en este territorio revelaron la necesidad de elaborar una herramienta informática para la gestión de la información para el desarrollo local.
- ✓ Las herramientas utilizadas en las distintas etapas de la implementación (WAMP5 con soporte para PHP, PostgreSQL y OOH como metodología específica para aplicaciones Web) son adecuadas en el desarrollo de este tipo de sistemas.
- ✓ A partir de la implementación del SGIDL, Ciego de Ávila cuenta con una herramienta capaz de gestionar la información necesaria para contribuir a la toma de decisiones por parte de los dirigentes de los gobiernos e instituciones para el desarrollo local.
- ✓ Los expertos consultados concuerdan en la efectividad y calidad del SGIDL y opinan que esta herramienta podría contribuir a aumentar notablemente la eficiencia en el proceso de gestión de la información para el desarrollo local en la Provincia Ciego de Ávila.

BIBLIOGRAFÍA

- [1] I. Gil Pechuan, "Sistemas y Tecnologías de la Información para la gestión," 2000.
- [2] L. Aja Quiroga, "Gestión de información, gestión del conocimiento y gestión de la calidad en las organizaciones," vol. 10, 2002.
- [3] C. Larrea Cepal, "El INFOPLÁN: Un sistema de información para el desarrollo local en el Ecuador.," 2002.
- [4] G. Ponjuán Dante, "Gestión de información en las organizaciones. Conceptos y Aplicaciones," 2004.
- [5] S. Artiles, "La gestión documental, de información y el conocimiento en la empresa: El caso de Cuba. ACIMED," 2009.
- [6] R. MY., "De la gestión de información a la gestión del conocimiento. Acimed.." vol. 2012, 2006.
- [7] R. G. Schroeder, "Administración de operaciones: toma de decisiones en la función de operaciones," 1983.
- [8] S. Valle Soler, "Fundamentos de sistemas de planificación de los recursos empresariales. Matanzas, Universidad Camilo Cienfuegos," 2007.
- [9] J. Salinas, "Innovación docente y uso de las TIC en la enseñanza universitaria," 2004.
- [10] C. Cachero, N. Koch, A. Kraus, and S. Melia, "Modeling web business processes with OO-H and UWE," 2003.
- [11] S. Bakken, "Manual de PHP," 2001.
- [12] H. M. Kienle, "It's About Time to Take JavaScript (More) Seriously," 2010.
- [13] R. Alarcón, "Diseño orientado a objetos con UML," 2000.
- [14] Y. Fan, "Cascading Style Sheets," 2011.
- [15] M. Cebrián de la Serna, *Creación de materiales para la innovación educativa con nuevas tecnologías*, 1998.
- [16] O. Pérez, "Bases de datos en PostgreSQL," 2003.
- [17] C. Casado Martínez, "Guía de aprendizaje Dreamweaver CS4.," 2011.
- [18] A. C. and Sierra, "Metodología de la investigación científica," 1995.

Recepción: 04 de diciembre de 2013

Aceptación: 11 de febrero de 2014

Publicación: 27 de marzo de 2014

PROPIUESTA DE PROCEDIMIENTO PARA REALIZAR PRUEBAS DE CAJA BLANCA A LAS APLICACIONES QUE SE DESARROLLAN EN LENGUAJE PYTHON

PROPOSED PROCEDURE FOR WHITE BOX TESTING APPLICATIONS DEVELOPED IN PYTHON LANGUAGE

Eduardo Salazar Martínez¹

1. Ingeniero en Ciencias Informáticas. Facultad Regional Granma de la Universidad de Granma. Cuba. E-mail: esalazar@grm.uci.cu

RESUMEN

El proceso de pruebas al software es uno de los aspectos fundamentales para medir el estado de calidad de un sistema informático e introducirlo satisfactoriamente en el mercado mundial. El objetivo del presente trabajo es elaborar la propuesta de un procedimiento para realizar pruebas, aplicando el método de Caja Blanca, a las aplicaciones que se desarrollan con lenguaje Python. En el mismo, se exponen las actividades a seguir por el Grupo de Calidad de la Facultad Regional Gramma, reflejando cada uno de los artefactos de entrada y salida que se generan, indicando cómo se utilizan y se completan.

ABSTRACT

The process of software testing is a fundamental aspect to measure the quality state of an informatics system so as to successfully introduce it into the market. The objective of this research is to present a proposal of procedure to apply tests as the beta testing (White Box) method to software developed in Python language. The main activities to follow by the Quality Group from the Gramma Regional Faculty show the entry and exit artifacts generated as well as indicate how they are used and complemented.

PALABRAS CLAVE

Artefactos. Grupo de Calidad. Procedimiento. Pruebas de Caja Blanca.

KEYWORDS

Artifacts. Quality Group. Procedure. White Box.

INTRODUCCIÓN

El acelerado avance de la ciencia y las Tecnologías de la Información y las Comunicaciones (TIC) ha traído como consecuencia que las empresas informáticas existentes se planteen el reto de brindar una respuesta rápida, eficaz y con calidad a los clientes que cada vez se vuelven más exigentes no sólo en cuanto al precio, sino también con la confiabilidad y seguridad que deben poseer los productos de software.

A partir del surgimiento de los grandes sistemas informáticos en el mundo, se ha incluido al mismo tiempo y por necesidad, el proceso de pruebas usado sistemáticamente junto al desarrollo de cada aplicación. Las pruebas de software (beta testing) son un conjunto de actividades que se llevan a cabo metódicamente, pueden planificarse por adelantado y ejecutarse una vez construido el código para la revisión final de las especificaciones del diseño y de la codificación del software. Las mismas, vistas desde el marco de un proceso de desarrollo de software, son los diferentes procesos que se ejecutan durante la realización de un producto, con el objetivo de asegurar efectividad, adecuada funcionalidad y su correcta validez.

Los procesos de pruebas de software, representa más de la mitad del coste de un programa, al requerir un tiempo similar al de la programación, esto conduce a un alto costo económico cuando no involucra vidas humanas, puesto que en este último caso el costo suele superar el 80 por ciento. Esta etapa sugiere ser más costosa que el propio proceso de desarrollo y diseño de sistemas. Así es que surge la necesidad de adoptar una alternativa adecuada, la cual exige herramientas y conocimientos destinados a realizar pruebas y la insistencia de comenzar a ejecutarlas desde la misma etapa de análisis y levantamiento de requerimientos. Desde un principio se puede incurrir en erróneas interpretaciones de las reglas del negocio, lo que finalmente tendrá como consecuencia la discrepancia entre lo que el cliente exige y el producto que se ha desarrollado.

Realizar las pruebas inequívocas de un programa implicaría ponerlo en todas las situaciones posibles, de esta forma se aseguraría que el mismo se encuentre libre de errores. También se considera importante la búsqueda de formas y métodos que garanticen resultados correctos. En la actualidad, un intento de mejoras lo constituye la automatización del proceso de pruebas.

Entre otras dificultades de los procesos de prueba de software se encuentran la falta de herramientas, debido fundamentalmente a su elevado precio o a que las existentes no se ajusten al propósito y/o entorno para el que se necesitan. También se le une la deficiencia de compatibilidad e interoperabilidad entre sistemas, la ausencia de un proceso básico de pruebas y el desconocimiento acerca de qué es lo que se debe probar. Sin menospreciar otros indicadores como la privación del uso de las herramientas de prueba que ya se posee, bien por su dificultad de uso o por falta de tiempo para aprender a manejarlas, por falta de soporte técnico, obsolescencia, por la formación inadecuada en el uso de las mismas, entre otros aspectos.

ESTADO DEL ARTE

Con el crecimiento acelerado de las tecnologías y la informática, la producción de software desempeña un papel importante, provocando a su vez una competencia en los sistemas, donde la calidad es fundamental para conseguir rentabilidad en la producción. La necesidad de realizar pruebas de calidad converge hacia el aseguramiento de la eficiencia del producto antes de salir al mercado.

Muchas empresas existentes dedicadas a la producción de software y gestión de la calidad, gozan de un alto prestigio y cuentan con sus propias estrategias de pruebas y herramientas de apoyo. Otras contratan a empresas e instituciones que se especializan en este proceso, entre ellas podemos mencionar a **GreenSQA** (Green Software Quality Assurance), con la misión de contribuir a la madurez de las empresas e industrias mediante el uso de servicios específicos de pruebas de software e implementación de sistemas de gestión de calidad, garantizando así procesos ágiles, confiables y eficientes.

Otro ejemplo es la empresa **SQS S.A.** (Software Quality System S.A.), que lleva acabo procesos de pruebas tanto en sus propias instalaciones como en las de sus clientes, asegurando la reducción de costes y el aumento de la calidad. Para ofrecer un mejor servicio a sus clientes, la compañía ha decidido organizar estos servicios de pruebas llevados a cabo en sus instalaciones bajo el nombre de **SOQUS, el Laboratorio de Testing de SQS**.

En este caso la **RCCS** (Red Colombiana de Calidad de Software), es un instrumento de gestión de conocimiento fundamentado en un modelo de ingeniería, que tiene como objetivo gestionar programas de apoyo a la implementación de modelos de software para fortalecer la industria nacional.

También al incremento de las tecnologías se vincula el desarrollo de aplicaciones, y en este caso nos interesan a aquellas que son creadas con la meta de poder automatizar el proceso de pruebas de código que se realizan sobre el software. Dentro del almacén de programas con este objetivo podemos encontrar las que están relacionadas con las pruebas de Caja Blanca. A continuación se mencionan algunas de estas herramientas.

- **JTest:** Es el primer sistema automático de búsqueda de errores en el código de programación para programadores de Java. Esta nueva tecnología desarrollada por la empresa **ParaSoft**, utiliza la Test Generation Technolgy (Tecnologías de Generación de Pruebas) para analizar programas en Java. Se trata de una herramienta que permite realizar análisis de código, pruebas unitarias automáticas y cobertura de código, así como generación dinámica de pruebas funcionales.

En el ámbito de los análisis dinámicos, **JTest** es capaz de generar automáticamente todas las pruebas unitarias que sean necesarias, teniendo en cuenta los parámetros de cobertura de código e intentando encontrar pruebas que deriven en errores de ejecución. Genera pruebas funcionales filtradas por las acciones y los datos, incluyendo peticiones *HTTP2* y *JDBC3*.

- **Insure++:** Es un entorno automatizado en una aplicación de herramientas de prueba C/C++ que detecta errores difíciles de localizar; como corrupción de la memoria, asignación de errores de memoria, errores de iniciación de variables, definición de conflictos entre variables, indicador de errores, errores de biblioteca, errores lógicos y errores en algoritmos. Sin embargo tiene licencia la cual hay que pagar un monto significativo cada cierto período de tiempo.
 - **BullseyeCoverage:** Es un analizador de código de cobertura para C ++ y C que indica cómo gran parte del código fuente se pone a prueba. Puede usar esta información para rápidamente centrar su esfuerzo de ensayo y determinar las áreas que necesitan ser revisadas. El código cobertura de análisis es útil durante la unidad de verificación, integración de pruebas y la liberación final. Permite crear código más fiable y ahorrar tiempo. Sin embargo la licencia se debe comprar cada cierto período de tiempo, según el cliente determine, oscilando de 500 euros a 1000 euros.
 - **LDRA:** Es una herramienta de control de calidad que ofrece un potente código fuente de pruebas y análisis para la validación y verificación de aplicaciones de software. Es importante cuando el software informático requiere ser fiable, robusto y libre de errores. Se trata de un potente y plenamente integrado suite de herramientas, que permite al software avanzado de técnicas de análisis que puedan aplicarse en las etapas claves del desarrollo del ciclo de vida. Sin embargo sus herramientas no contienen la comprobación de estándares de codificación y para su empleo hay que comprarla a precios estimados.
 - **Logiscope TestChecker:** Aplicación para la representación gráfica de cobertura del código fuente. Evalúa el nivel de cobertura del código, el usuario debe de comprar la licencia por período de tiempos, no realiza evaluaciones al código de C Sharp y no cuenta con la verificación de estándares.
 - **CMT++:** Es una herramienta para la medida de la complejidad para C/C ++, la misma es fácil de utilizar para ambos lenguajes. También el código en ensamblador se puede medir con esta herramienta. CMT++ está destinado para organizaciones desarrolladoras de software que se esfuerzan por un proceso de desarrollo productivo resultante en productos de alta calidad.
- Ayuda a estimar el mantenimiento general del código base y a localizar fácilmente las partes complejas de éste. Se pueden evaluar por separado: poniendo especial atención a las pruebas, o tal vez rediseñándolas. Se puede utilizar CMT++ también para medir la cantidad de código que se tiene: líneas físicas, líneas de comentarios, líneas de programa, declaraciones. Es una herramienta que a pesar de sus características no es gratuita y la documentación para su aprendizaje se encuentra en otros idiomas exceptuando el castellano.
- **CTC++:** Con esta herramienta ocurre lo mismos inconvenientes que con CMT++, aunque no se debe de dejar de mencionar que Testwell CTC++ (Test Analizador de Cobertura para C y C++) es una herramienta de cobertura código/prueba potente y fácil de usar, la cual muestra las partes del código que han sido ejecutadas

(probadas). La herramienta analiza todos los niveles de cobertura requeridos en proyectos “críticos” y ayuda a garantizar una mayor calidad del código. Testwell CTC++ puede utilizarse para obtener las certificaciones en la industria automotriz, aérea y médica.

Visto el estudio y análisis en el marco internacional de las empresas desarrolladoras de software y gestión de la calidad, y ejemplos de aplicaciones existentes que se suelen emplear para garantizar la mayor calidad posible de los sistemas sometiéndolos a procesos de pruebas, la idea de alcanzar una empresa o sistema que asegure la calidad en el desarrollo de software se ha venido fortaleciendo en Cuba de forma continua. Existen empresas en el territorio que se inclinan hacia el mismo objetivo, como **CITMATEL** (Empresa de Tecnologías de la Información y Servicios Telemáticos Avanzados) que se distingue entre las entidades cubanas por su excelencia y creciente proyección hacia el mercado externo e interno, con una amplia diversidad de productos y servicios integrales de alto valor agregado. Entre las principales líneas de trabajo se destaca el desarrollo de sistemas dirigidos a automatizar la gestión en empresas de todo tipo.

Otro ejemplo es **ALBET** (Albet Ingeniería y Sistemas), cuyo origen y desarrollo se vincula estrechamente a la Universidad de Ciencias Informáticas (UCI), modelo de universidad productiva que agrupa una multitud de profesionales, técnicos y estudiantes. No podemos dejar de mencionar que la UCI cuenta también con un Centro para la Certificación de la Calidad de Software CALISOFT, unido al Departamento de Producción de Software. El centro cuenta con trabajos de diploma investigativos que se han realizado referentes a los temas sobre procedimientos generales de pruebas de Caja Blanca y procesos de pruebas referidos al método de Caja Negra. Existen en algunos proyectos productivos en la UCI, la utilización de herramientas que aplican diferentes pruebas de Caja Negra, probando la funcionalidad del software y en la minoría (prácticamente ninguno) se aplica el método de Caja Blanca de forma manual a un pequeño fragmento de código. El Grupo de Calidad de la Facultad Regional de la UCI en Granma, también participa en la realización de pruebas de software, aplicando generalmente pruebas de Caja Negra y de Carga. Por consiguiente, existe la carencia de procedimientos de pruebas de Caja Blanca o empleo de aplicaciones de apoyo que permitan la automatización de estos procesos en la universidad.

CALIDAD DE SOFTWARE

La calidad de software es un problema actual que afecta tanto a los productores de software como a los clientes. Con el aumento de la informatización a escala mundial la demanda de software crece exponencialmente y los desarrolladores le han brindado poco interés a la calidad de sus productos. Sigue que muchas veces los clientes reciben el software cuando se han violado las etapas de pruebas.

La calidad del software puede definirse de muchas maneras. Una de las más limitadas, conocida como “calidad pequeña”, define la calidad como la ausencia de defectos [4]. Para evaluarla de esta forma se emplean procedimientos estadísticos a partir de las tendencias de aparición de fallas durante la prueba de software.

“Existen estándares industriales que marcan aceptabilidad cuando se estima el número de defectos residuales en 0.02 defectos por millar de líneas de código y aún menos.” [5]

“Otros enfoques de calidad consideran diversos factores, entre ellos la confiabilidad” [6]. Existe una larga tradición de estudio de la confiabilidad que se asocia estadísticamente con el comportamiento del software.

Existen varias formas de definir la confiabilidad, en unos casos se considera tiempo de operación y en otros la variedad de usos propuestos. Una definición más reciente, plantea que: “La confiabilidad es la probabilidad de operación exitosa de un programa dado, en un intervalo de tiempo, en un ambiente específico”. [6]

Obteniendo la calidad requerida en el software, se logra reducir su número de errores o eliminarlos completamente, se alcanza una mayor fiabilidad para las funciones que debe realizar el mismo, mayor eficiencia e integridad de los datos así como flexibilidad y reusabilidad.

“La calidad de software es una actividad de protección que se aplica a lo largo de todo el proceso de Ingeniería del Software. Esta engloba los siguientes aspectos:” [6]

- Un enfoque de gestión de calidad.
- Tecnología de Ingeniería del Software efectiva (métodos y herramientas).
- Revisiones técnicas formales que se aplican durante el proceso del software.
- Una estrategia de prueba multiescala.
- El control de la documentación del software y de los cambios realizados.
- Un procedimiento que asegure un ajuste a los estándares de desarrollo del software.
- Mecanismos de medición y de generación de informes.

“La calidad debe ser especificada, planificada, administrada, medida y certificada. Esto implica una visión integral que arroja la comprobación del software, con el fin de lograr un mayor grado de satisfacción y confianza del cliente hacia la organización productora de

software. Constituye entonces las pruebas de los software, tarea de alta prioridad para las empresas productoras". [2].

Analizando los concepto expuesto sobre la calidad por varios autores y haciendo una inclinación por este último expresado por Pressman gracias a que se considera como uno de los más completos y acordes al objeto de estudio, también cabe decir que la calidad es considerada una disciplina integral y a la vez una cualidad indisoluble del software para su comprobación, muy ligada a las empresas productoras como tarea fundamental en sus procesos de pruebas.

PRUEBAS DE SOFTWARE

Unas de las vías más importantes para determinar el estado de la calidad de un producto de software es el proceso de pruebas. Estas están dirigidas a componentes del sistema en su totalidad, con el objetivo de medir el grado en que cumple con los requerimientos. En ellas se usan casos de prueba, especificados de forma estructurada mediante técnicas. Sus objetivos, métodos y técnicas usadas se describen en el plan de prueba.

La prueba es una actividad fundamental en muchos procesos de desarrollo, incluyendo el del software. Estas permiten detectar la presencia de errores que pudieran generar las entradas o salidas de datos y comportamientos inapropiados durante su ejecución. Un concepto más específico dado por algunos desarrolladores de software es:

“Cualquier intento de demostrar que el software tiene propiedades por debajo de la calidad requerida”. [7].

De acuerdo a la IEEE [8] el concepto de prueba se define como:

“Una actividad en la cual un sistema o componente es ejecutado bajo condiciones específicas, se observan o almacenan los resultados y se realiza una evaluación de algún aspecto del sistema o componente”. [9].

Otro concepto importante a tomar en consideración es el emitido por Pressman en su edición de 1998, que plantea lo siguiente:

“La prueba del software es un elemento crítico para la garantía de calidad del software y representa una revisión de las especificaciones, del diseño y de la codificación”. [9].

Teniendo en cuenta las definiciones anteriores se puede concluir que la prueba de software es una actividad en la cual el sistema es ejecutado bajo condiciones específicas para demostrar que no tiene la madurez necesaria para ser implantado. Dentro de las actividades que se practican para obtener un software con la madurez necesaria están:

- Revisiones: consiste en que cada integrante del equipo de desarrollo revisa el producto que va generando.
- Inspecciones: revisión de cada producto por parte de colegas.
- Validaciones: es el cliente quien revisa el producto para decir si cumple con sus necesidades.

Esta definición implica que se considera una prueba exitosa si no se demuestran deficiencias en el software. Las fallas pueden ser en el código o en el modelado, en dependencia del tipo de pruebas que se le apliquen al software.

Se distinguen pruebas técnicas y pruebas funcionales. Las pruebas técnicas son la responsabilidad de los ingenieros de software que han desarrollado el producto, pero estos ingenieros en ocasiones deben hacerse cargo de las pruebas funcionales.

En proyectos a gran escala las pruebas funcionales son la responsabilidad de un equipo de pruebas, formado por uno o varios técnicos, un coordinador de pruebas y un gestor de pruebas o de calidad.

CARACTERÍSTICAS GENERALES DE LA ESTRATEGIA DE PRUEBA

Al aplicarles las pruebas al software se deben seguir un conjunto de estrategias para lograr que éstas se hagan en el menor tiempo posible y con la calidad requerida, además de garantizar que arrojen los resultados esperados.

Dentro de las características generales de la estrategia de prueba se encuentran. [2]

1. La prueba comienza en el nivel de módulo y trabaja “hacia fuera”, hacia la integración completa del sistema completo.
2. En diferentes puntos es adecuada la utilización de técnicas de prueba distintas.
3. La prueba la lleva a cabo el que desarrolla el software; y para grandes proyectos, un grupo de prueba independiente.
4. La prueba y la depuración son actividades diferentes, pero la depuración puede entrar en cualquier estrategia de prueba.

Hay dos estrategias generales para la prueba de software: las estrategias de pruebas de especificación (Caja Negra) y pruebas de código (Caja Blanca).

MÉTODOS DE PRUEBAS

Existen diversos métodos para realizar las pruebas de software, entre las más importantes se encuentran la prueba de Caja Blanca, prueba de Caja Negra y prueba de la Estructura de Control.

El uso de la prueba de Caja Blanca es mejor para verificar que se recorran todos los caminos y detectar un mayor número de errores. La Caja Negra brinda la posibilidad de cubrir la mayor parte de las combinaciones de entradas y lograr así un juego de pruebas más eficaz.

Las pruebas mencionadas permiten probar cada una de las condiciones existentes en el programa, identificar claramente las entradas, salidas y estudiar las relaciones que existen entre ellas, permitiendo así maximizar la calidad de las pruebas y en dependencia del resultado se constará con un sistema más estable y confiable.

PRUEBA DE ESPECICACIÓN (CAJA NEGRA)

Pruebas de Caja Negra: También suelen ser llamadas funcionales y basadas en especificaciones. En ellas se pretende examinar el programa en busca de que cuente con las funcionalidades que debe tener y como lleva a cabo las mismas, analizando siempre los resultados que devuelve y probando todas las entradas en sus valores válidos e inválidos.

Al ejecutar las pruebas de Caja Negra se desarrollan casos de prueba reales para cada condición o combinación de condiciones y se analizan los resultados que arroja el sistema para cada uno de los casos. En esta estrategia se verifica el programa considerándolo una caja negra. Las pruebas no se hacen en base al código, sino a la interfaz. No importa que se cubran todas las rutas dentro del programa, lo importante es probar todas las entradas en sus valores válidos e inválidos y lograr que el sistema tenga una interfaz amigable.

LIMITACIONES

Lograr una buena cobertura con pruebas de caja negra es un objetivo deseable; pero no suficiente a todos los efectos. Un programa puede pasar con holgura millones de pruebas de especificación y sin embargo tener defectos internos que surgen en el momento más inoportuno.

Por ejemplo, una computadora que contenga el virus Viernes-13 puede estar pasando pruebas de caja negra durante años y años. Sólo falla si es viernes y es día 13; pero ¿a quién se le iba a ocurrir hacer esa prueba? [11]

Las pruebas de caja negra nos convencen de que un programa realizar bien sus funcionalidades programadas, pero no de que haga (además) otras cosas menos aceptables.

PRUEBA DE CÓDIGO (CAJA BLANCA)

Pruebas de Caja Blanca: También suelen ser llamadas estructurales o de cobertura lógica. En ellas se pretende investigar sobre la estructura interna del código, exceptuando detalles referidos a datos de entrada o salida, para probar la lógica del programa desde el punto de vista algorítmico. Realizan un seguimiento del código fuente según se va ejecutando los casos de prueba, determinándose de manera concreta las instrucciones, bloques, etc. que han sido ejecutados por los casos de prueba.

En las pruebas de Caja Blanca se desarrollan casos de prueba que produzcan la ejecución de cada posible ruta del programa o módulo, considerándose una ruta como una combinación específica de condiciones manejadas por un programa.

Hay que señalar que no todos los errores de software se pueden descubrir verificando todas las rutas de un programa, hay errores que se descubren al integrar unidades del sistema y pueden existir errores que no tengan relación con el código específicamente.

CARACTERÍSTICAS DE LAS PRUEBAS DE CAJA BLANCA

En las pruebas de Caja Blanca, se pretende indagar sobre la estructura interna del código, omitiendo detalles referidos a datos de entrada o salida. Su objetivo principal es probar la lógica del programa desde el punto de vista algorítmico.

Éstas se basan en el diseño de Casos de Prueba que usa la estructura de control del diseño procedimental para derivarlos. Mediante las pruebas de Caja Blanca el ingeniero de software puede obtener Casos de Prueba que: [11]

- Garanticen que se ejercent por lo menos una vez todos los caminos independientes de cada módulo, programa o método.
- Ejercent todas las decisiones lógicas en las vertientes verdadera y falsa.
- Ejercent todos los bucles en sus límites operacionales.
- Ejercent las estructuras internas de datos para asegurar su validez.

Las pruebas de Caja Blanca son consideradas entre las más importantes que se aplican a los sistemas, con la que se obtienen como resultados la disminución en un gran por ciento el número de errores existentes en el software y por ende una mayor calidad y confiabilidad en la codificación.

TIPOS DE PRUEBAS DE CAJA BLANCA

De estructura de datos locales:

Se centran en el estudio de las variables del programa. Busca que toda variable esté declarada y que no existan con el mismo nombre, ni declaradas local y globalmente, que haya referencias a todas las variables y para cada variable, analiza su comportamiento en comparaciones.

De cobertura lógica:

1. *De Cobertura de Sentencias:* Comprueba que todas las sentencias se ejecuten al menos una vez.
2. *De Cobertura de Decisión:* Ejecuta casos de prueba de modo que cada decisión se pruebe al menos una vez a Verdadero (True) y otra a Falso (False).
3. *De Cobertura de Condición:* Ejecuta un caso de prueba a True y otro a False por cada condición, teniendo en cuenta que una decisión puede estar formada por varias condiciones.
4. *De Cobertura de Condición/Decisión:* Se realizan las pruebas de cobertura de condición y las de decisión a la vez.
5. *De Condición Múltiple:* Cada decisión multicondición se traduce a una condición simple, aplicando posteriormente la cobertura de decisión.
6. *De Cobertura de Caminos:* Se escriben casos de prueba suficientes para que se ejecuten todos los caminos de un programa. Entendiéndose camino como una secuencia de sentencias encadenadas desde la entrada del programa hasta su salida.

[11]

PRUEBA DEL CAMINO BÁSICO

Buscando una mejor comprensión de los contenidos, se hace importante definir primeramente algunos conceptos fundamentales:

Camino: “Secuencia de todas las instrucciones de un programa de principio a fin”. [2] Un camino se puede definir como la ruta de secuencias que se siguen dentro del código de fuente de un programa, un ejemplo es, desde la entrada de valores al sistema hasta la devolución de resultados que arroja, respetando la estructura de código.

Camino Básico: Es una técnica de prueba de Caja Blanca que permite obtener una medida de complejidad lógica para generar un conjunto básico de caminos que se ejecutan por lo menos una vez durante la ejecución del programa. [12]

La prueba del camino básico es una técnica de pruebas de Caja Blanca propuesta por Tom McCabe. Esta técnica permite obtener una medida de la complejidad lógica de un diseño y usar esta medida como guía para la definición de un conjunto básico. La idea es derivar casos de prueba a partir de un conjunto dado de caminos independientes por los cuales puede circular el flujo de control.

Camino independiente: “Es cualquier camino del programa que incluye nuevas instrucciones de un proceso o una nueva condición”. [12]

El conjunto de caminos independientes se obtiene construyendo el Grafo de Flujo asociado y se calcula su complejidad ciclomática. Por último se diseñan los casos de prueba y se ejecutan los mismos.

Complejidad: Es proporcional al número de errores en un segmento de código. “Entre más complejo, más susceptible a errores”. [13] Se relaciona con el esfuerzo requerido para probar. “Entre más complejo, mayor atención para probar”. [13]

Complejidad ciclomática: Es la “medida de la complejidad lógica de un módulo “G” y el esfuerzo mínimo necesario para calificarlo. Es el número de rutas lineales independientes de un módulo “G”, por lo tanto es el número mínimo de rutas que deben probarse”. [13]

Esta técnica ofrece una gran ventaja con respecto a las otras técnicas, ya que el número mínimo requerido de pruebas se sabe por adelantado y por tanto el proceso de prueba se puede planear y supervisar en mayor detalle.

Los pasos a seguir para aplicar esta técnica son:

- Representar el programa en un grafo de flujo.
- Calcular la complejidad ciclomática.
- Determinar el conjunto básico de caminos independientes.
- Derivar los casos de prueba que fuerzan la ejecución de cada camino.

A continuación, se detallan cada uno de estos pasos.

1) REPRESENTACIÓN DE UN GRAFO DE FLUJO:

El grafo de flujo se utiliza para representar el flujo de control lógico de un programa. Este emplea los tres elementos siguientes:

- **Nodos:** Representan cero, una o varias sentencias en secuencia. Cada uno comprende como máximo una sentencia de decisión (bifurcación).
- **Aristas:** Líneas que unen dos nodos.
- **Regiones:** Áreas delimitadas por aristas y nodos. Cuando se contabilizan las regiones de un programa debe incluirse el área externa como una región más.

Así, cada construcción lógica de un programa tiene una representación. La Figura 1 muestra un grafo de flujo del diagrama de módulos correspondiente. Nótese cómo la estructura principal corresponde a un **while** y dentro del bucle se encuentran anidados dos constructores **if** [14]

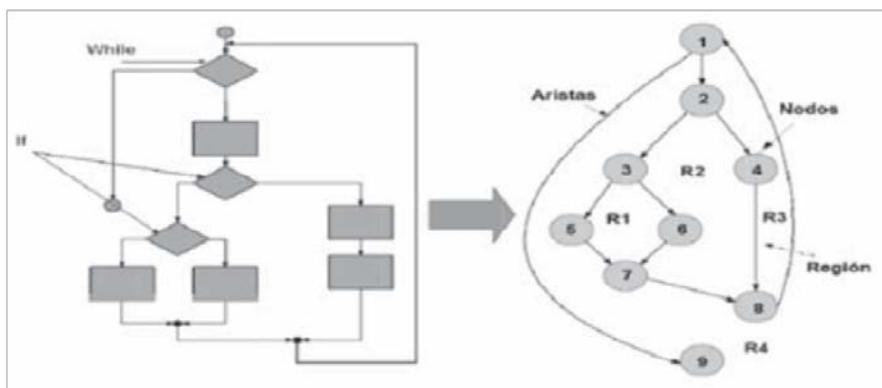


Figura 1. Ejemplo de grafo de flujo correspondiente a un diagrama de módulos

2) CALCULAR LA COMPLEJIDAD CICLOMÁTICA:

La complejidad ciclomática es una métrica del software que proporciona una medida cuantitativa de la complejidad lógica de un programa. Se basa en la representación gráfica del flujo de control del programa, el análisis desprende una medida cuantitativa de la dificultad de prueba y una indicación de la fiabilidad final. Cuando se utiliza en el contexto del método de prueba del camino básico, el valor calculado como complejidad ciclomática define el número de pruebas que se deben realizar para asegurar que se ejecute cada sentencia al menos una vez.

Métrica: “Medida estadística que se aplica a todos los aspectos de calidad de software, los cuales deben ser medidos desde diferentes puntos de vista”. [13] La métrica se define como término para valorar en una medida la calidad de software considerando diferentes puntos necesario a evaluar.

Es una de las métricas de software más ampliamente aceptada, ya que ha sido creada para ser independiente del lenguaje. Se han realizado investigaciones y ensayos, en los cuales se ha medido un gran número de programas, a modo de establecer rangos de complejidad que

ayuden al ingeniero de software a determinar la estabilidad y riesgo inherente de un programa. La medida resultante puede ser utilizada en el desarrollo, mantenimiento y reingeniería para estimar el riesgo, costo y estabilidad.

Estos estudios indicaron la existencia de distintas relaciones entre la métrica de McCabe y el número de errores existentes en el código fuente, así como el tiempo requerido para encontrar y corregir esos errores. “Se puede comparar la complejidad ciclomática contra un conjunto de valores límites como se observa en la Tabla 1”. [13].

Complejidad Ciclomática	Evaluación del Riesgo
1 - 10	Programa Simple, sin mucho riesgo.
11 - 20	Más complejo, riesgo moderado.
21 - 50	Complejo, Programa de alto riesgo.
50	Programa no testeable, muy alto riesgo.

Tabla 1. Complejidad ciclomática vs Evaluación de riesgo

El ámbito más común de utilización para probar módulos unitarios es la prueba estructural (Caja Blanca). McCabe también expone que se puede utilizar la complejidad ciclomática para dar una indicación cuantitativa del tamaño máximo de un módulo. A partir del análisis de muchos proyectos se encontró que un valor 10 es un límite superior práctico para el tamaño de un módulo.

Cuando la complejidad supera este valor se hace muy difícil probarlo, entenderlo y modificarlo. La limitación deliberada de la complejidad en todas las fases del desarrollo ayuda a evitar los problemas asociados a proyectos de alta complejidad. El límite propuesto por McCabe sin embargo es fuente de polémicas. Algunas organizaciones han utilizado el valor 15 con bastante éxito. No obstante, un valor superior a 10 debería ser reservado para proyectos que tengan ventajas operacionales con respecto a proyectos típicos.

La complejidad ciclomática puede ser aplicada en varias áreas incluyendo: [15]

- *Análisis de Riesgo en desarrollo de código:* Mientras el código está en desarrollo, su complejidad puede ser medida para estimar el riesgo inherente.
- *Análisis de riesgo de cambio durante la fase de mantenimiento:* La complejidad del código tiende a incrementarse a medida que es mantenido durante el tiempo. Midiendo la complejidad antes y después de un cambio propuesto, puede ayudar a decidir cómo minimizar el riesgo del cambio.
- *Planificación de Pruebas:* El análisis matemático ha demostrado que la complejidad ciclomática indica el número exacto de casos de prueba necesarios para probar cada punto de decisión en un programa.
- *Reingeniería:* Provee conocimiento de la estructura del código operacional de un sistema. El riesgo involucrado en la reingeniería de una pieza de código está relacionado con su complejidad.

¿Por qué es tan importante?

Permite apreciar la calidad del diseño de software de una manera rápida y con independencia del tamaño de la aplicación. Es una medida esencial cuando se necesita “tomar la temperatura” de un diseño de software de un tamaño considerable (más si se está realizando una auditoría externa y no se conocía de antes la aplicación), y que se puede obtener fácilmente de manera automatizada. También se ha utilizado para planificar proyectos de mejora de grandes productos de software, para priorizar las partes del diseño.

Por otro lado, en muchas ocasiones, es base para calcular el valor de las mejoras del diseño, o el valor que aporta introducir un patrón o buena práctica. Además, da el número de casos de prueba unitarios básicos para obtener una cobertura del 100 por ciento, y un aproximado al grado de comprensibilidad.

¿Cómo sería su cálculo?

“Existen varias formas de calcular la complejidad ciclomática (CC) de un programa a partir de un grafo de flujo: [14]

- $CC = \text{arcos} - \text{nodos} + 2$
- $CC = \text{número de nodos de decisión} + 1$
- $CC = \text{número de regiones}^*$.

Esta complejidad ciclomática determina el número de casos de prueba que deben ejecutarse para garantizar que todas las sentencias de un programa se han ejecutado al menos una vez, y que cada condición se habrá ejecutado en sus vertientes verdadera y falsa. A continuación se expone cómo se identifican estos caminos.

3) DETERMINAR EL CONJUNTO DE CAMINOS BÁSICOS INDEPENDIENTES:

Camino linealmente independiente de otros: Introduce por lo menos un nuevo conjunto de sentencias de proceso o una nueva condición.

Un camino independiente es cualquier camino del programa que introduce, por lo menos, un nuevo conjunto de sentencias de proceso o una condición, respecto a los caminos existentes. En términos del diagrama de flujo, un camino independiente está constituido por lo menos por una arista que no haya sido recorrida anteriormente a la definición del camino. En la identificación de los distintos caminos de un programa para probar se debe tener en cuenta que cada nuevo camino debe tener el mínimo número de sentencias nuevas o condiciones nuevas respecto a los que ya existen. De esta manera se intenta que el proceso de depuración sea más sencillo.

El conjunto de caminos independientes de un grafo no es único. No obstante, a continuación, se muestran algunas heurísticas para identificar estos caminos: [14]

- Elegir un camino principal que represente una función válida que no sea un tratamiento de error. Debe intentar elegirse el camino que atraviese el máximo número de decisiones en el grafo.
- Identificar el segundo camino mediante la localización de la primera decisión en el camino de la línea básica alternando su resultado mientras se mantiene el máximo de número de decisiones originales del camino inicial.
- Identificar un tercer camino, colocando la primera decisión en su valor original a la vez que se altera la segunda decisión del camino básico, mientras se intenta mantener el resto de decisiones originales.
- Continuar el proceso hasta haber conseguido tratar todas las decisiones, intentando mantener como en su origen el resto de ellas.

Heurística: La búsqueda a ciegas es aquella donde no existe ninguna información para decidir que nodo expandir, no se conoce la cantidad de pasos o el costo del camino desde el estado actual hasta el objetivo. También se denomina búsqueda no informada. En el otro caso, cuando existe información para decidir, la búsqueda se denomina informada o heurística. [16]

Este método permite obtener de la complejidad ciclomática, caminos independientes cubriendo el criterio de cobertura de decisión y sentencia.

Por ejemplo, para el grafo de la Figura 1 los cuatro posibles caminos independientes generados serían:

Camino 1: 1-9
 Camino 2: 1-2-4-8-1-9
 Camino 3: 1-2-3-5-7-8-1-9
 Camino 4: 1-2-3-6-7-8-1-9

4) DERIVAR LOS CASOS DE PRUEBA:

El último paso es construir los casos de prueba que fuerzan la ejecución de cada camino. Una forma de representar el conjunto de casos de prueba sería llenar la “Tabla 1.2” [14] que se muestra a continuación.

Número del Camino	Caso de Prueba	Resultado Esperado

Tabla 1.2. Posible representación de casos de prueba para pruebas estructurales.

PLAN DE PRUEBAS

“El propósito del plan de pruebas es dejar de forma explícita el alcance, el enfoque, los recursos requeridos, el calendario, los responsables y el manejo de riesgos de un proceso de pruebas”. [17]

Está constituido por un conjunto de pruebas. Cada prueba debe:

- Dejar claro qué tipo de propiedades se quieren probar (corrección, robustez, fiabilidad, amigabilidad, etc.).
- Dejar claro cómo se mide el resultado.
- Especificar en qué consiste la prueba (hasta el último detalle de cómo se ejecuta).
- Definir cuál es el resultado que se espera (identificación, tolerancia,...). ¿Cómo se decide que el resultado es acorde con lo esperado?

Las pruebas carecen de utilidad, tanto, si no se sabe exactamente lo que se quiere probar, si no se está claro cómo se prueba, o si el análisis del resultado se hace a simple vista.

Estas mismas ideas se suelen agrupar diciendo que un caso de prueba consta de tres bloques de información:

- El propósito de la prueba.
- Los pasos de ejecución de la prueba.
- El resultado que se espera.

Todos y cada uno de esos puntos deben quedar perfectamente documentados. El plan de pruebas señala el enfoque, los recursos y el esquema de actividades de prueba, así como los elementos a probar, las características, las actividades de prueba, el personal responsable y los riesgos.

EL PROCESO DE PRUEBAS

El proceso de pruebas de un software consta de varias etapas dentro de ellas las más importantes son:

- *Inspección del análisis:* Se verifica si se cometieron errores o falla en la etapa de análisis.
- *Inspección del diseño:* Se comprueba el diseño y se trata de hallarle defectos.
- *Inspección del código:* Se observa el entendimiento y facilidad del código.
- *Pruebas unitarias:* Se debe probar cada método de las clases implementadas por separado.
- *Pruebas de integración:* Se prueban todas las clases, verificando que compaginen entre sí.
- *Pruebas de validación de requerimientos:* Validan que se cumple con todos los requerimientos exigidos por el cliente.

- *Pruebas de sistema:* Ejecutar el programa para verificar si cumple con los requisitos exigidos.

PRUEBAS DE UNIDAD

Se comprueban los módulos cada uno por separado, buscando errores en el funcionamiento de ese módulo como sistema independiente. Estas pruebas deben ser hechas por el diseñador y el programador del módulo.

PRUEBAS DE SISTEMA

Su objetivo es la comprobación del sistema global, se realizan pruebas de tres tipos distintos:

- Seguridad: protección en aplicaciones especialmente sensibles a entradas no deseadas.
- Resistencia: se prueba la robustez del sistema frente al mal uso de la aplicación por parte de ciertos usuarios.
- Rendimiento: eficiencia medida en velocidad de proceso y recursos consumidos.

PRUEBAS DE INTEGRIDAD

Las pruebas de integración se llevan a cabo durante la construcción del sistema, involucran a un número creciente de módulos y terminan probando el sistema como conjunto. Estas pruebas se pueden plantear desde un punto de vista estructural o funcional.

Las pruebas estructurales de integración son similares a las pruebas de caja blanca; pero trabajan a un nivel conceptual superior. En lugar de referirse a sentencias del lenguaje, se refiere a llamadas entre módulos. Se trata de identificar todos los posibles esquemas de llamadas y ejercitártolos para lograr una buena cobertura de segmentos o de ramas.

Las pruebas funcionales de integración son similares a las pruebas de caja negra. Aquí trataremos de encontrar fallos en la respuesta de un módulo cuando su operación depende de los servicios prestados por otro(s) módulo(s). Según nos vamos acercando al sistema total, estas pruebas se van basando más y más en la especificación de requisitos del usuario.

Las pruebas finales de integración cubren todo el sistema y pretenden cubrir plenamente la especificación de requisitos del usuario. Además, a estas alturas ya suele estar disponible el manual de usuario, que también se utiliza para realizar pruebas hasta lograr una cobertura aceptable.

PRUEBAS DE ACEPTACIÓN

El uso de cualquier producto de software tiene que estar justificado por las ventajas que ofrece. Sin embargo, antes de su puesta en marcha es muy difícil determinar si sus ventajas realmente justifican su uso. El mejor instrumento para esta determinación es la llamada “prueba de aceptación”. En esta prueba se evalúa el grado de calidad del software con relación a todos los aspectos relevantes para que el uso del producto se justifique.

Eliminar la influencia de conflictos de intereses, y para que sea lo más objetiva posible, la prueba de aceptación no debería ser responsabilidad de los ingenieros de software que han desarrollado el producto.

En la preparación, ejecución y evaluación de las pruebas de aceptación no se requiere de conocimientos informáticos. Sin embargo, un conocimiento amplio de métodos y técnicas de prueba y de la gestión de la calidad en general facilita esta labor.

La persona adecuada (o el equipo adecuado) para llevar a cabo la prueba de aceptación disponen de estos conocimientos y además son capaces de interpretar los requerimientos especificados por los futuros usuarios del sistema de software en cuestión.

Estas pruebas las realiza el cliente. Son básicamente pruebas funcionales sobre el sistema completo, y buscan una cobertura de la especificación de requisitos y del manual del usuario. Estas pruebas no se realizan durante el desarrollo, pues sería impresentable al cliente; sino que se realizan sobre el producto terminado e integrado o pudiera ser una versión del producto o una iteración funcional pactada previamente con el cliente.

La experiencia muestra que aún después del más cuidadoso proceso de pruebas por parte del desarrollador, quedan una serie de errores que sólo aparecen cuando el cliente comienza a usarlo.

Sea como sea, el cliente siempre tiene razón. Decir que los requisitos no estaban claros, o que el manual es ambiguo puede salvar la cara; pero ciertamente no deja satisfecho al cliente.

Una prueba de aceptación puede ir desde un informal caso de prueba hasta la ejecución sistemática de una serie de pruebas bien planificadas. De hecho, las pruebas de aceptación pueden tener lugar a lo largo de semanas o meses, descubriendo así errores latentes o escondidos que pueden ir degradando el funcionamiento del sistema. Estas pruebas son muy importantes, ya que definen las nuevas fases del proyecto como el despliegue y mantenimiento.

Se emplean dos técnicas para las pruebas de aceptación:

- **La prueba alfa.**

Se lleva a cabo, por un cliente, en el lugar de desarrollo y en un entorno controlado. Se usa el software de forma natural con el desarrollador como observador del usuario. Para que tengan validez, se debe primero crear un ambiente con las mismas condiciones que se encontrarán en las instalaciones del cliente. Una vez logrado esto, se procede a realizar las pruebas y a documentar los resultados. [3]

- **La prueba beta.**

Se lleva a cabo por los usuarios finales del software y se realiza en el entorno de los clientes. A diferencia de la prueba alfa, el desarrollador no está presente normalmente. Así, la prueba beta es una evaluación “en tiempo real” del software en un entorno no planificado por el

desarrollador. El cliente registra todos los problemas (reales o imaginarios) que encuentra durante la prueba beta e informa a intervalos regulares al desarrollador.

Como resultado de los problemas informados durante la prueba beta, el desarrollador del software lleva a cabo modificaciones y así prepara una versión del producto de software para todos los cliente donde se despliegue el producto. [3]

FLUJO ACTUAL DE LOS PROCESOS

El proceso de control de calidad antes, durante y después de la implementación de un producto de software es una tarea bastante complicada incluso para los expertos en el tema, ya que nunca se tiene la última palabra a cerca de estos factores y cada situación que se presenta puede resultar novedosa y problemática simultáneamente.

En la Facultad Regional de la Universidad de las Ciencias Informáticas en Granma (FRG-UCI) se producen sistemas destinados al Sector de Cultura, en su mayoría con dimensiones grandes y compuestos por diferentes líneas de trabajo como los Sistemas de Gestión de Información, Realidad Virtual y Desarrollo de Aplicaciones para dispositivos móviles.

El proceso de pruebas en estos sistemas se realiza mensualmente por el Grupo de Calidad (GC) de la FRG-UCI, y una vez estén terminados los productos, el personal implementador también forma parte de este proceso. Las pruebas que se realizan con mayor frecuencia son las Pruebas de Estrés, Pruebas de Funcionalidades y a la Documentación por lo general. El jefe del equipo de prueba recibe todo el sistema y la documentación a probar, por medios de la herramienta de control de versiones Bazaar, y a medida que se realizan las revisiones, manualmente se van conformando los registros de no conformidades guiado por una plantilla, estos últimos los recibe el líder de proyecto para hacer cambios a favor de superar la calidad de la aplicación y nuevamente someterla a otra revisión, y así se repite el proceso hasta que el producto quede con el mínimo número de errores posible.

Las pruebas de Caja Blanca todavía no se comprenden dentro de las que se practican el GC a los productos en desarrollo. Cada proyecto revisa el código fuente usando la técnica más conveniente o que mejor conozca, y los pocos que documentan los resultados de las pruebas no lo hacen debidamente.

CONCLUSIONES

Los conceptos principales sobre la calidad y pruebas de software logran describir aspectos relacionados con los procesos de pruebas de sistemas identificando las estrategias de pruebas. Las búsquedas de los tipos de pruebas ayudan a resumir sus características y a describir los flujos actuales del proceso de calidad.

REFERENCIAS BIBLIOGRAFICAS

- CIG_LABS. *The Home of Groundbreaking Software Quality Management Research*, [en línea]. [consultado 10/02/2012]. Disponible: www.cigallabs.com/reso/definitions/software_testing.html.
- Darias Pérez Darling, *Análisis y Diseño de Componentes para Pruebas de Caja Blanca*. UCI, 2008. Cuidad de La Habana, Cap. 1. 14, 15 p.
- Díaz Yanersy, MOLINA Yenisel. *Diseño de una aplicación para el Seguimiento de Errores de los productos software de la Facultad 7*. UCI. 2007. Cuidad de la Habana.
- Esteban Bello Rafael, Zenaida García Lorenzo Zoila, M. García Moreno María, Reynoso Lobato Antonio, *Aplicaciones de la Inteligencia Artificial*. Primera edición. 2002, p. 13.
- Fernández Peña J. M. IPN México. *Pruebas de integración para componentes de software*, Marzo 2002.
- Fernández, Giovanny. *Estándar codificación DOTNET*. España. 2005. (2008). 20 p.
- IEEE Std 1995, *Metrics*, IEEE, 1991.
- Juristo Natalia, MORENO Ana M., VEGAS Sira. *Técnicas de Evaluación de Software*. 2006. (2008). 131 p. / UNIVERSIDAD SIMON BOLIVAR. Prueba repetible y mantenible. [en línea]. Bitácora de Prueba. Ingeniería de Software 3. Enero-Marzo 2001. [en línea]. [consultado 12/02/2012]. Disponible: <http://www.tuobra.unam.mx/publicadas/040803214240.html>.
- Kan, S.H. 1995: *Metrics and models in software quality engineering*, AddisonWesley, Reading, Ma., USA, 1995.
- Marcelo Rizzo, Francisco. Reportes Técnicos, Complejidad Ciclomática. ITBA. CAPI. 8 p.
- Márquez Alpizar Yaimí, VALDEZ HECHAVARRIA Yenni. *Procedimiento General de Pruebas de Caja Blanca aplicando la técnica de Camino Básico*. UCI, 2007. Cuidad de La Habana. Cap. 1. 20, 21 p.
- Meyer, B. 1997: *Object oriented software construction*, 2nd. De., Upper Saddle River, Prentice Hall, 1997.
- Polo Usaola, Dr. Macario. *Curso de doctorado sobre Proceso software y gestión del conocimiento. Pruebas del Software*. Departamento de Tecnologías y Sistemas de Información. Ciudad Real. 2006. (2008). 46 p.
- Roger S. Pressman, R. *Can Internetbased Applications Be Engineered?* in IEEE Software, September/October IEEE Press, 104110, 1998.

- Roger S. Pressman: *Software Engineering: A Practitioner's Approach* (European Adaptation), McGrawHill. 2000. ISBN: 0077096770.
- T. J. McCabe, Structured testing: a testing methodology the cyclomatic complexity metric, Technical Report NIST 500-225, 1996.
- Teruel Alejandro. [en línea]. [citado 16/02/2012]. Disponible: <http://www.ldc.usb.ve/~teruel/ci4713/clases2001/pruebasRep.html#bitacora>.
- Yamaura, Tsuneo 1998: How to design practical test cases, IEEE Software. Vol. 15, n6, november/december 1998, pp 3036.

Recepción: 07 de mayo de 2014

Aceptación: 02 de junio de 2014

Publicación: 26 de junio de 2014

APLICACIONES INFORMÁTICAS PARA LA GESTIÓN ECONÓMICA FINANCIERA Y EL ASEGURAMIENTO DE LA UNIVERSIDAD DE LA HABANA

**COMPUTER APPLICATIONS FOR FINANCIAL MANAGEMENT
AND FINANCIAL ASSURANCE OF THE UNIVERSITY OF LA
HABANA**

Lic. Viviana Duro Novoa¹

1. Profesora Instructora, Asesora Informática Vicerrectoría Económica de la Universidad de La Habana. E-mail: viviana.duro@rect.uh.cu

RESUMEN

El presente trabajo muestra una propuesta de Informatización para la gestión económica financiera (GEF) de la Universidad de La Habana. Aborda el prototipo de una infraestructura para la gestión a través de la Web, que garantizará un fácil acceso a la información integrada por cada uno de los niveles de interés, una actividad económica más eficiente, mayor nivel de seguridad e integridad de la información y una toma de decisiones más certera en el plano económico.

Como resultados se presentan las aplicaciones: Portal de Tramites Digitales, Sistema de Solicitudes de Servicio, Sistema de Gestión del Mantenimiento, Sistema de Gestión del Transporte, Sistema de Control Energético, Sistema de Gestión de Almacenes, Sistema de Reportes para la Toma de Decisiones, Portal de Atención al Usuario y la Web de la Gestión Económica Financiera como plataforma que integra las aplicaciones antes mencionadas. Existen además otros proyectos de sistemas en fase de desarrollo.

ABSTRACT

This paper presents a proposal of Computerization for economic and financial management (EFM) of the University of Havana. It addresses the prototype of a management infrastructure through the Web, which will ensure an easy access to information integrated by each of the levels of interest, a more efficient economic activity, higher level of security and integrity of information and a decision making more accurate in the economic sphere.

As result these applications are presented: Procedures Digital Portal, Service Request System, Maintenance Management System, Transport Management System, Energy Control System, Warehouse Management System, Reporting System for Decision Making, Customer Care Portal and web of Financial and Economic Management platform that integrates the aforementioned applications. There are other systems projects under development.

PALABRAS CLAVE

Gestión económico-financiera, toma de decisiones, bases de datos, plataforma, aplicaciones.

KEY WORDS

Economic and financial management, decision making, database, platform, applications.

INTRODUCCIÓN

Desde mediados del año 2003 se vienen produciendo cambios y modificaciones por parte de los organismos rectores de la economía cubana. En consecuencia, fue necesario, oportuno y conveniente adecuar las actividades económicas en el Ministerio de Educación Superior (MES), y a su vez las de las universidades adscritas a éste, lo que quedó recogido en el Nuevo Modelo de Gestión Económico-Financiera (NMGEF)(1), como solución integral y sistemática a las deficiencias e insuficiencias de la gestión hasta el momento.

El NMGEF presupone una forma perfeccionada del desempeño en la actividad económico financiera en el MES, en permanente evolución y transformación. No es más que un proceso de sistematización y reordenamiento que tiene en cuenta en su diseño todas las orientaciones, indicaciones y resoluciones de los organismos rectores de la economía cubana. Está conformado por sistemas y subsistemas que agrupan todas las técnicas y conocimientos requeridos para el desempeño de las tareas que tienen como propósito el cumplimiento de la misión de las áreas económicas en nuestras universidades.

El Subsistema de Informática y Automatización que tiene como objetivo general “la elevación de la eficacia, eficiencia y efectividad en la planeación, asignación y el control de los recursos financieros, humanos y materiales; así como en los diversos procesos de toma de decisiones en la esfera de la actividad económica y de aseguramiento de la Educación Superior Cubana, mediante el diseño y aplicación de un nuevo modelo de gestión de la información económica, basado en la informatización de los procesos de generación, comunicación, procesamiento y almacenaje de información”.

Y plantea como parte de sus objetivos específicos:

- Garantizar la total fiabilidad de la información de trabajo y reflejar los hechos económicos durante la captación, procesamiento, almacenamiento, distribución y consolidación a los diferentes niveles del Centro.
- Agilizar el procesamiento de información y obtención de los indicadores que se definan.
- Facilitar un trabajo más integral y corporativo entre todos los eslabones del Área de Economía.
- Brindar niveles de seguridad informática y protección del acceso a la información existente de acuerdo a lo establecido por los organismos rectores de esta actividad.
- Posibilitar la gestión documentaria moderna en las oficinas del Área de Economía.

Desde los primeros momentos en que se comenzó a trabajar en el Programa de Perfeccionamiento se acometieron tareas de vital importancia relacionadas con la informatización del proceso económico-contable de las entidades, siempre en el marco de la Estrategia Maestra de Informatización del Ministerio de Educación Superior (2), que plantea entre sus metas:

- Elevar de la cultura de la informatización en todos los niveles.
- Asimilar, introducir, y actualizar las mejores tecnologías informáticas de equipamiento, información y comunicaciones al menor costo posible.
- Aplicar las tecnologías de la información y las comunicaciones (TIC) en todas las actividades principales de la Educación Superior y potenciar nodos de excelencia.
- Generar nuevas tecnologías de la informática, información y las comunicaciones, lograr mayor incorporación de profesores, investigadores y estudiantes a este propósito.

En este sentido y tomando en consideración lo planeado en el NMGEF, se crea dentro del aparato económico de la universidad un grupo de informatización que se encarga, en particular, de la implantación, mantenimiento y desarrollo de sistemas informáticos en estas áreas y actividades y del diseño del desarrollo estratégico del uso de las tecnologías para la gestión económica y financiera en la UH.

Teniendo como resultado el mejoramiento de la infraestructura tecnológica, la puesta en funcionamiento en todas las UP adscritas a la UH, la más moderna versión del Sistema de Gestión Contable Assets Premium¹, y el desarrollo de una primera versión de web para encontrar información estática extraída de dicho sistema y referida al trabajo de las áreas económicas.

Además, está implementado en la UH por parte de la Dirección de Informatización de la universidad el Sistema de Gestión de la Nueva Universidad (SIGENU)² para el registro de la matrícula, de cuyos reportes estadísticos se sirve el aparato económico para el desarrollo de su actividad cuyo uso está aún en proceso de perfeccionamiento.

Lograr la gestión online de todos los procesos de apoyo de la UH tendría, sin duda, un impacto trascendental. En general, se evidencia la necesidad de un impulso en este sentido.

¹ Sistema desarrollado por D'Marco y comercializado en Cuba por INFOMASTER que permite registrar las operaciones de compras, ventas, producción, taller, inventario, finanzas, contabilidad, presupuesto, activos fijos, útiles y herramientas y recursos humanos. Sistema integral, todos sus módulos trabajan en estrecha relación, dejando el registro de las transacciones efectuadas. Aplicación cliente servidor y motor de Bases de Datos SQL Server.

² Sistema creado por el Ministerio de Educación Superior con el fin de ser una herramienta que permita la gestión de toda la información académica vinculada con la Educación Superior en Cuba

DESARROLLO

El grupo de Informatización de la Gestión Económica, con el fin de crear plataformas tecnológicas integradoras que permitan planificar, gestionar, dirigir y controlar online los procesos de la gestión económica y financiera de la UH para que constituyan herramientas para la toma de decisiones en el año 2009, se propone una estrategia (3) de trabajo cuyos objetivos fundamentales fueron:

- La Planeación, asignación y el control de los recursos financieros, humanos y materiales.
- Constituir una plataforma para la gestión de procesos y la toma de decisiones en la Gestión Universitaria.
- Ser herramienta de comunicación y generar mejoras continuas en la satisfacción de intereses de la comunidad universitaria.

Partiendo de la Identificación de necesidades de información, el estudio de los flujos de información y el análisis y rediseño de los procesos y procedimientos de gestión interna y servicio susceptibles a optimizar mediante soluciones tecnológicas y con el fin de incrementar la transparencia en la gestión, eliminar el uso de papeles en todos los procesos en que sea posible, evitar los desplazamientos de directivos y funcionarios, elevar la cultura económica de la comunidad universitaria, potenciar las capacidades en el desarrollo de la actividad fundamental y la implantación de sistemas de calidad en los servicios económicos y la actividad de apoyo, medir y comprobar el cumplimiento de indicadores, se llevaron a cabo desarrollos de aplicaciones informáticas en ambientes web de trabajo.

Se utiliza para el desarrollo de estas plataformas las siguientes tecnologías:

- Servidor Web Apache (v2.2.21)³
- Motor de Base de Datos MySQL (v5.5.16)⁴
- Para el modelado de la Base de Datos se utilizó MySQL WorkBench (v5.1), el cual permite de forma visual crear modelos entidad-relacional y posteriormente sincronizarlos con la base de datos.
- Se utiliza PHP⁵ como lenguaje de programación de lado del servidor.

³ El servidor HTTP Apache es un servidor web de código abierto para plataformas Unix (BSD, GNU/Linux, etc) Microsoft Windows, Macintosh y otras. Es usado principalmente para enviar páginas web estáticas y dinámicas en la World Wide Web.

⁴ Es un sistema de gestión de base de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones en el mundo. Actualmente este sistema de gestión de base de código abierto es muy utilizado en aplicaciones web y en plataformas Linux/Windows-Apache-MySQL-PHP/Perl/Python

- CodeIgniter (v2.1) como framework⁶ de desarrollo utilizando el patrón de diseño Modelo Vista Controlador⁷. Este expone librerías y “helpers” que facilitan el desarrollo ágil de aplicaciones web.
- Es utilizado extensamente a lo largo de los proyectos jQuery y jQueryUI para un mejor manejo de la sintaxis JavaScript⁸ y para mejorar la experiencia de usuario mostrando controles avanzados para interaccionar con la aplicación.
- La plataforma se encuentra enlazada con varias bases de datos sobre SQL Server 2008 del Sistema de Gestión Contable Assets Premium de las cuales extrae datos frecuentemente.
- Se ha utilizado GIT como sistema de control de versiones durante todas las fases de desarrollos de las plataformas.

Aplicaciones desarrolladas:

- Portal de Tramites Digitales.
- Sistema de Solicitudes de Servicio.
- Sistema de Gestión del Mantenimiento.
- Sistema de Gestión del Transporte.
- Sistema de Control Energético.
- Sistema de Gestión de Almacenes.
- Sistema de Reportes para la Toma de Decisiones.
- Portal de Atención al Usuario.
- Web de la Gestión Económica Financiera.

⁵ PHP (en inglés, PHP Hypertext Pre-processor): Publicado bajo la licencia de software libre PHP License, es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas con acceso a información almacenada en una base de datos.

⁶ Los framework para aplicaciones web están diseñados para apoyar el desarrollo de sitios web dinámicos, aplicaciones y servicios web. Agilizan actividades comunes del desarrollo web, proporcionando bibliotecas para acceder a bases de datos, estructuras para plantillas y gestión de sesiones, entre otras utilidades. En la mayoría de los casos, facilitan la reutilización de código.

⁷ Arquitectura Model View Controller (MVC) o Modelo Vista Controlador. Este es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. Lo cual es considerado una buena práctica que facilita la reutilización de código y múltiples presentaciones para una misma aplicación.

⁸ Es un lenguaje de programación, orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico. Todos los navegadores modernos interpretan el código JavaScript integrado en las páginas web. JavaScript se interpreta en el agente de usuario, al mismo tiempo que las sentencias van descargándose junto con el código HTML.

Aplicaciones en fase de desarrollo como son:

- Sistema para el Control Financiero de la Universidad.
- Sistema de Planificación.

PORTAL DE TRÁMITES DIGITALES

Esta aplicación se desarrolla con el fin de resolver los problemas existentes en el llenado de los modelos para trámites: autorización de cobro, conteo del 10% de activos fijos tangibles, movimiento de activos fijos tangibles (traslados, altas, bajas, otros) y solicitudes de materiales, ya que la información necesaria se obtiene a través de una conexión directa a los servidores de bases de datos del sistema contable ASSETS ubicados en cada una de las áreas económicas, dando la opción de enviar e imprimir el modelo listo.

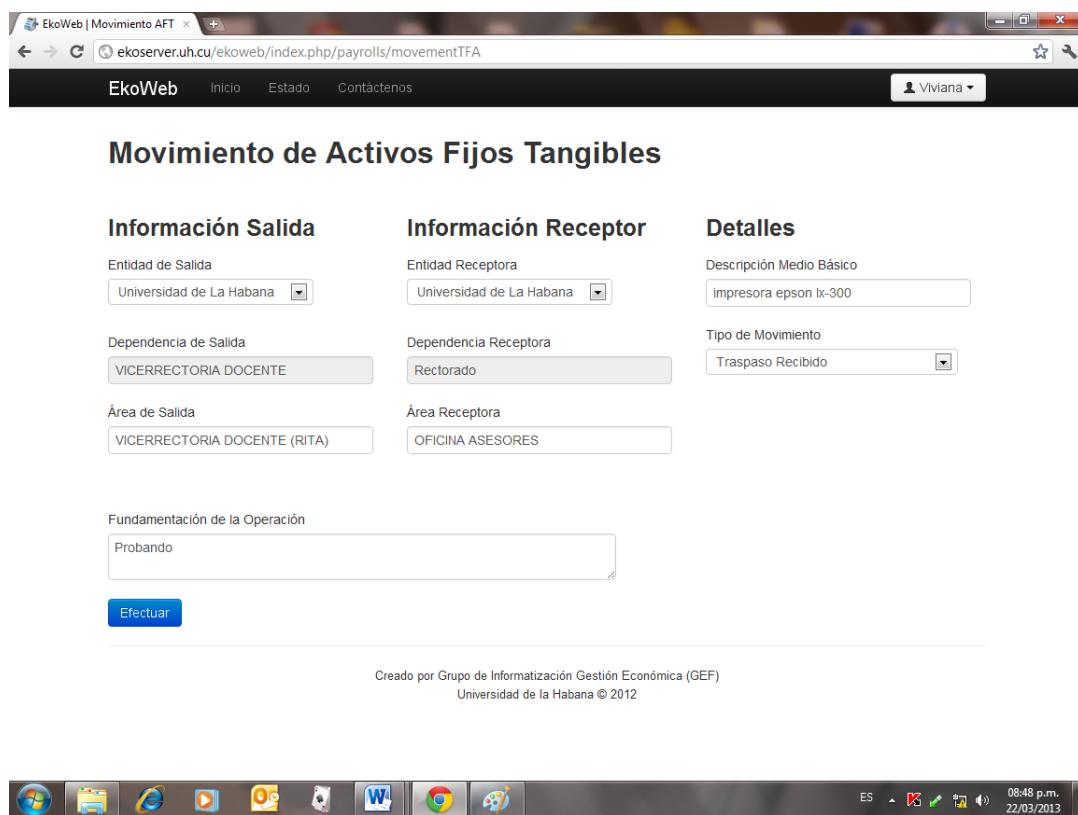


Fig. 1. Formulario para el Movimiento de Activos Fijos Tangibles

SISTEMA DE SOLICITUDES DE SERVICIO

Este sistema se desarrolla con el objetivo de dar la posibilidad a las áreas universitarias de realizar las solicitudes de servicios que requieran: reparación de averías, impresión y transportación permitiéndoles conocer en todo momento el estado de su solicitud, el proceso comienza con la solicitud del usuario y no se da por terminado hasta tanto el área no evalúa el servicio recibido de satisfactorio o insatisfactorio dando de este modo los primeros pasos para gestionar la calidad.



Fig. 2. Formulario de Solicitud de Servicio de Impresión

GESTIÓN DE REPARACIÓN Y MANTENIMIENTO ON-LINE

Dirección de Obras y Mantenimiento

Para el desarrollo de este sistema se tuvieron en cuenta tres niveles fundamentales de necesidades de información.

La información para la Dirección, para la que se elaboran informes sobre:

- Costes de Mantenimiento (origenes, composición, distribución por centros de costo, etc.)
- Objetivos y seguimiento presupuestario.
- Resultados económicos (seguimiento y análisis para generar actuaciones).

La información para las operaciones y la gestión diaria:

- Órdenes de trabajo (conocimiento de los parámetros fundamentales de la ejecución de los trabajos).
- Disponibilidad y Eficacia (conocimiento del impacto de la gestión diaria sobre los indicadores fundamentales).
- Análisis de averías y equipos críticos (conocimiento y explotación de la base de datos de averías para la determinación de las actuaciones preferentes de mantenimiento de cara a mejorar la situación y criticidad de los equipos).

La información para el puesto de trabajo:

- Sistema de incentivos.
- Necesidades de formación (detección de las necesidades de formación del personal de cara a la introducción de procesos de mejora continua).
- Control de resultados para la contribución a canalizar las mejoras.
- Realimentación de la información.

Se informatizaron los documentos: Reporte de Avería, Programa diario de Trabajo (visitas programadas, fechas de entrada y salida); Órdenes de Trabajo, permitiendo conocer cualquier momento la situación de un trabajo en cuanto a: fechas de visita, técnico responsable de materiales (disponibilidad en almacén, compra a terceros, medios propios o recuperados), mano de obra (fuerza de trabajo necesaria disponible) y equipamiento.

Permite llevar los Controles Técnicos (control de horas x hombre, control de cargas de trabajo, control de trabajos más importantes (prioridades), control de trabajos pendientes, control de averías y causas, control de locales con averías repetitivas), los controles económicos (resumen de control de costos a nivel de centro de costo comparado con presupuesto, control de centros de costo desviados hasta nivel petición de trabajo) y consultar fichas históricas (listados de trabajos pendientes, listados de trabajos terminados, tiempos de respuesta, evaluación de los trabajos realizados por parte de las áreas).

Se relaciona con otras aplicaciones informáticas:

- **Interface con Sistema de Gestión Contable Assets Modulo Inventarios:** A través del Código de Almacén se traen al Sistema Mantenimiento todos los datos de la Gestión del Aprovisionamiento relativo a la avería en cuestión.
- **Interface con Sistema de Gestión de Almacenes:** A través del N° de Reporte y el nivel de Defectos se llegará a: productos no existentes, unidad de medida y cantidad necesaria (y todos los datos asociados). Genera Plan de Compra, en la propia aplicación Web.
- **Interface con Gestión Financiera (en fase de desarrollo):** A través del Código del Centro de Costo se llegará a los gastos por área, por epígrafes y partidas contemplando vales de Salida de Almacén y pagos a terceros.

Mantenimiento



Existe un faltante de materiales para su problema de Electricidad.
Se necesitan 2 'iPhone', si su área posee los mismos por medios propios, por favor notifique de su existencia.

[Notificar existencia](#)
[Mostrar listado materiales](#)



Se ha emitido una orden de comienzo y finalización para su problema de Refrigeración. Las obras de mantenimiento comenzarán en '2012-07-20' y terminarán en '2012-07-21'

[Ver Solicitud de Materiales](#)



Se ha emitido una orden de comienzo y finalización para su problema de Carpintería. Las obras de mantenimiento comenzarán en '2012-07-25' y terminarán en '2012-07-26'

[Ver Solicitud de Materiales](#)

Fig. 3. Ventana de Sistema de Gestión del Mantenimiento.

APLICACIONES INFORMÁTICAS PARA LA GESTIÓN ECONÓMICA FINANCIERA Y EL ASEGURAMIENTO DE LA
UNIVERSIDAD DE LA HABANA

Viviana Duro Novoa

123

GESTIÓN DEL CONTROL ENERGÉTICO DE LA UH

Dirección de Servicios Generales

El problema a resolver con esta aplicación es controlar el consumo eléctrico de la Universidad, llevar las estadísticas para poder elaborar los planes de consumo y, a través del análisis de los datos, contribuir a disminuir los gastos de electricidad en la Universidad.

Parte de realizar un levantamiento de cargas teniendo en cuenta el consumo diario y el tiempo de uso de todos los Activos Fijos Tangibles, registrados en las bases de datos del sistema contable Assests de las Direcciones de Economía. Usando ya esta información los especialistas de la Dirección de Servicios Generales pueden introducir el plan para cada una de las áreas, modificarlo y se le da la posibilidad al resto de los usuarios de consultarlos, por otra parte los administradores de las áreas quienes antes reportaban el consumo a través de la vía telefónica, proceden a introducir la lectura diaria de los metros contadores.

La aplicación cuenta con tres secciones: **Introducir Datos, Utilidades y Planes de Consumo**.

Dentro de la primera sección **Introducir Datos**, existen dos acciones: Insertar y Actualizar el consumo de los activos fijos. Estas acciones solo pueden ser realizadas por especialistas. En ambas básicamente se hace lo mismo, primero se escoge un área de responsabilidad de las que pertenecen al centro de costo del usuario, y se listan sus activos fijos que consumen electricidad, y a partir de este listado se puede elegir cualquiera de éstos y el usuario introduce los valores del tiempo que se mantiene encendido el equipo y lo que consume teóricamente, la única diferencia es que con la primera opción solo aparecen listados elementos a los cuales no se les ha introducido nunca un valor de consumo eléctrico en la base de datos de la aplicación, para facilitar a los usuarios el levantamiento inicial de los valores.

En la segunda sección, **Utilidades**, existen dos acciones: reportes ejecutivos y calcular gastos.

En los reportes ejecutivos valiéndose de la retroalimentación brindada por la introducción de las lecturas de los metros, los planes y el consumo teórico de las áreas, se elaboran informes sobre las áreas que más consumen, las de menos consumo, se establecen comparaciones de lo real consumido con respecto al plan aprobado, entre otros.

En la acción de calcular gastos, se calcula el gasto en CUP a partir de un valor que se introduzca de kilowatts consumidos basándose en el costo para el sector residencial de la electricidad.

En la tercera sección se agrupan las acciones correspondientes al trabajo con los planes de consumo y el consumo eléctrico en general. La primera de las acciones nos permite introducir el consumo, o sea la lectura de un metro contador en específico mediante un listado de todos los metros contadores.

En la segunda etiquetada con consumo eléctrico, se muestra un gráfico donde se refleja la relación del consumo del área a la que pertenece el usuario con respecto a la lectura total del

metro contador a la que pertenece. Lo logramos separando la carga teórica del área, lograda con el levantamiento de los datos hecho en la primera sección de la aplicación.

En la tercera etiquetada con Plan mensual, se muestra una vista con precisamente, el plan desglosado por meses, donde el especialista de la Dirección de Servicios Generales puede editar cada uno de ellos.



Fig. 4. Sistema para el control del consumo energético de la Universidad

GESTIÓN DEL TRANSPORTE

Dirección de Servicios Generales

Esta plataforma comprende el parque de vehículos, conductores, las hojas de rutas, el control del combustible asignado, los mantenimientos y los modelos establecidos para el control de esta actividad.

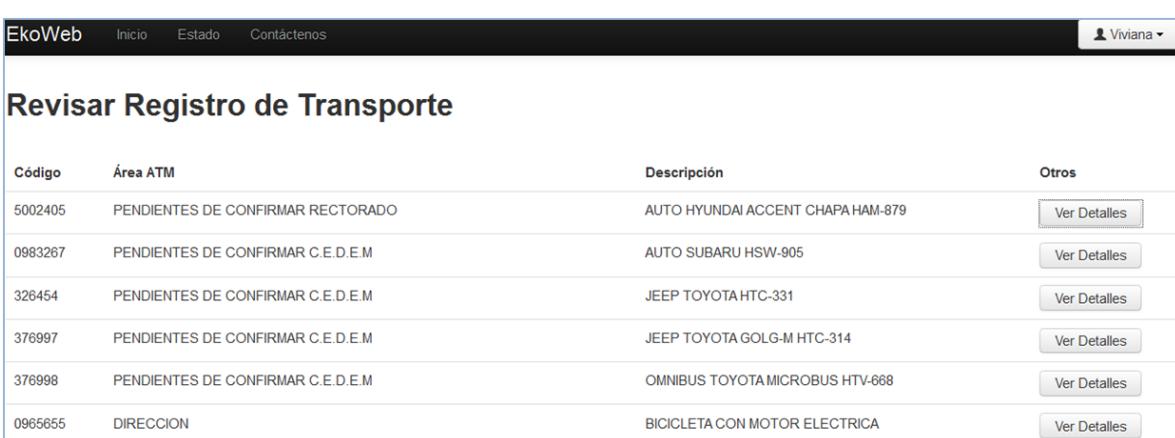
El Registro de Vehículos de la Universidad se obtiene del listado de Activos Fijos Tangibles de la Base de Datos de la Dirección Económica de la Unidad Presupuestada de Aseguramiento a la Universidad de La Habana, que están categorizados como medios de transporte (Número de Inventario, Área, Descripción) y se complementa con el resto de la información que registra el Departamento de Transporte relacionada con los aspectos técnicos de los vehículos: chapa, color del vehículo, número del motor, número de la carrocería, número de circulación, número de licencia operativa (para camiones), año de fabricación, capacidad de carga (personas o toneladas, en dependencia del vehículo), tipo de vehículo, ring de los neumáticos, fecha de vencimiento de la circulación, fecha de vencimiento de la licencia operativa (para camiones), fecha de vencimiento del somatón y foto.

La información de los conductores es extraída de la Base de Datos de Recursos Humanos: código, nombre y apellidos, carnet de identidad, dirección particular, área a la que pertenece, cargo que ocupa, fecha de alta a la universidad y, además, recoge datos esenciales para el funcionamiento del departamento de transporte, como número de licencia de conducción, fecha de alta como conductor, acumulación de puntos de la licencia de conducción, chapa del vehículo asignado, tipo de conductor y foto.

Con esa información la aplicación permite procesar las hojas de ruta y llenar los modelos D.E.G-008 y Control de Entrega de Combustible, posibilita planificar el mantenimiento de los vehículos para cada mes y realizar la distribución de combustible de todos los vehículos de la entidad por parte del Director General de la UPA y el Director de Servicios Generales, generando el modelo establecido. Siempre haciéndole llegar la notificación a las áreas que corresponda con cada operación.

Brinda reportes ejecutivos como: el listado de los vehículos cuyo período de vencimiento del somatón sea menor de 2 meses, los vehículos cuyo período de vencimiento de la licencia operativa sea menor de 2, consultar la disponibilidad de combustible actual de cada vehículo.

Permitir la consulta tanto del Jefe de Transporte o el Jefe de la Piquera como de los Jefes de Área de un resumen de incidencias de los vehículos que controlan, ya sea en el consumo de combustible como de las irregularidades que puedan darse en las hojas de ruta y la notificación de los mantenimientos planificados.



Código	Área ATM	Descripción	Otros
5002405	PENDIENTES DE CONFIRMAR RECTORADO	AUTO HYUNDAI ACCENT CHAPA HAM-879	<button>Ver Detalles</button>
0983267	PENDIENTES DE CONFIRMAR C.E.D.E.M	AUTO SUBARU HSW-905	<button>Ver Detalles</button>
326454	PENDIENTES DE CONFIRMAR C.E.D.E.M	JEEP TOYOTA HTC-331	<button>Ver Detalles</button>
376997	PENDIENTES DE CONFIRMAR C.E.D.E.M	JEEP TOYOTA GOLG-M HTC-314	<button>Ver Detalles</button>
376998	PENDIENTES DE CONFIRMAR C.E.D.E.M	OMNIBUS TOYOTA MICROBUS HTV-668	<button>Ver Detalles</button>
0965655	DIRECCION	BICICLETA CON MOTOR ELECTRICA	<button>Ver Detalles</button>

Fig. 5. Ventana de consulta Registro de Vehículos

GESTIÓN DE ALMACENES

Dirección de Abastecimiento Técnico Material

Esta aplicación comprende el registro de inventarios, salidas al costo, solicitudes de materiales, distribuciones materiales, gestión de compras. Se vincula con el sistema de control de inventario que posee la universidad y le permite a las áreas chequear "on line" sus inventarios.

Se basa en la formulación de pedidos por parte de las áreas, sobre la base de una determinada oferta. Genera una oferta "virtual" que permitirá una racionalización de los inventarios y una reducción de los gastos en las gestiones.

Las solicitudes de servicio y los mínimos de inventarios permiten generar el plan de compra, pudiendo hacer un análisis de los productos que mas rotan. Se vincula además con el sistema de gestión financiera en el desarrollo. El esquema de su funcionamiento se puede ver en la gráfica siguiente:

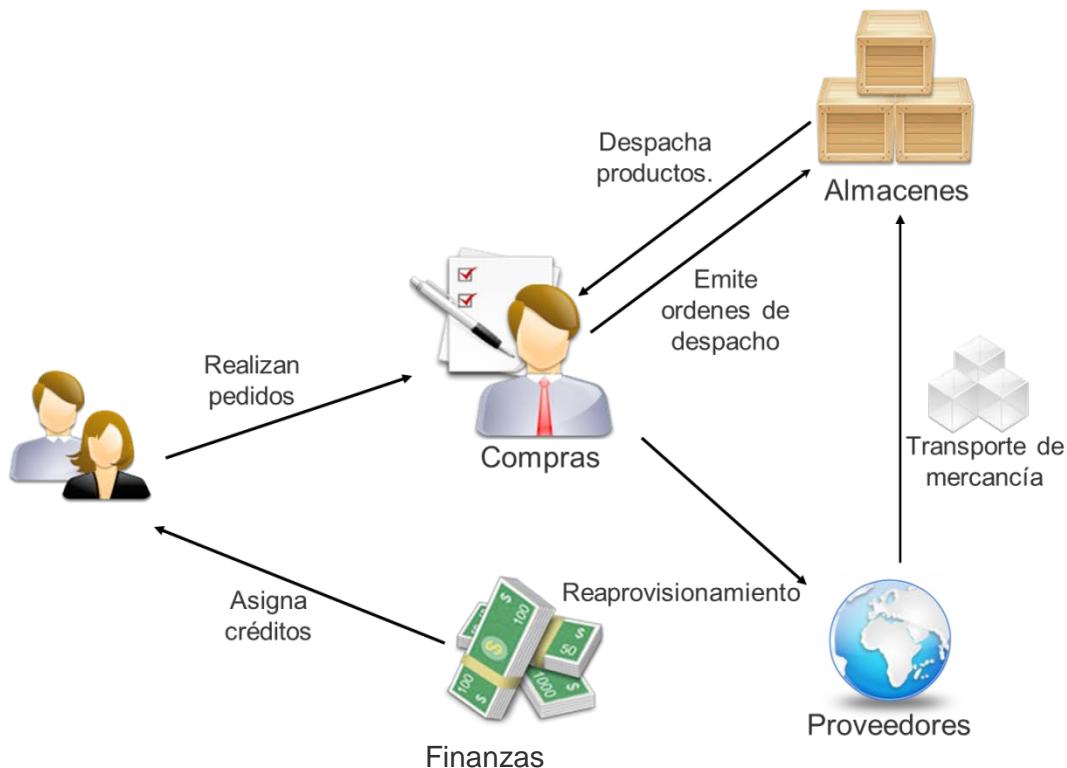


Fig. 6. Gráfico de Adquisición de productos áreas universitarias, reabastecimiento almacenes.

Fuente: Elaboración propia

WEB DE LA GESTIÓN ECONÓMICA FINANCIERA

Esta aplicación se desarrolla a fin de resolver uno de los requerimientos para disponer de un entorno de acceso a la información de la GEF y el aseguramiento donde se integran todos los sistemas explicados en el artículo con anterioridad, funcionando como la cara de las estructuras que atienden cada uno de estos procesos de apoyo de la universidad, desde esta aplicación se establecen los niveles de acceso a la información y los roles para cada tipo de usuario.

Roles	Acceso a la información
Rector	Toda la información y gestión.
Vicerrector	La información y gestión referente a su vicerrectoría y las Direcciones que se le subordinan.
Directores	La información y gestión de su dirección.

Decanos	La información y gestión de su facultad.
Administradores	La información y gestión del área que administran.
Especialistas	La información y gestión de los servicios que prestan, reportes.

Fig. 7. Tabla de roles de usuarios y oportunidades de información. **Fuente:** Elaboración propia.

Forman parte de la misma el Portal de Atención al Usuario con el Servicio “Contáctenos”, reportes de problemas con cualquiera de las aplicaciones, quejas o sugerencias sobre los sistemas y el centro de notificaciones además de la aplicación para los reportes para la toma de decisiones.

Esta plataforma permite planificar, gestionar, dirigir, y controlar online los procesos de la gestión económica y financiera de la UH, y que constituye una herramienta para la toma de decisiones.



Ventana incial Ekoweb



Portal de Atencion al Usuario

Centro de Notificaciones

Su solicitud de impresión para el trabajo titulado "Web Performance" ha sido procesado.

Solicitudes de Impresión

Mantenimiento



Movimiento Activos Fijos de Dir. Atención Eventos

Estadística	Cantidad
Total Movimientos Realizados	79
Trámites Recibidos	14
Alta	0
Trámites Efectuados	0
Períodos	0
Trámites Internos	35
Actas Ocasional	0
Bajos	30

Reportes Toma de decisiones

Fig. 8. Vistas EkoWeb

Además de todas las aplicaciones que ha desarrollado el grupo de informatización de la Gestión Económica, también como parte de su estrategia de desarrollo para el uso de todas estas aplicaciones que se han ido implantando, se han dictado cursos presenciales de capacitación a los usuarios y se han diseñado cursos en la plataforma moodle de e-learning, que están disponibles desde la intranet de la UH al igual que un Foro abierto para la aclaración de las dudas que pueda tener la comunidad universitaria sobre los procesos llevados a cabo por las estructuras de la Gestión Económica Financiera y de Aseguramiento de la Universidad.



Fig. 9. Curso Sistema Assets en plataforma Moodle.

VALORACIÓN DEL TRABAJO

La valoración de los trabajos en el marco de la Informatización de la Gestión Económica Financiera y el Aseguramiento está dada fundamentalmente por:

- Acceso rápido y seguro a la información.
- Implicación en la toma de decisiones de los niveles de dirección.
- Ahorro de tiempo.
- Ventajas en precisión.

Uno de los beneficios radica en la automatización de las tareas que hasta el momento se realizaban manualmente, a partir de grandes volúmenes de información económica, contable y financiera, que evidentemente redundaría en un determinado número de errores al manipular la misma.

Se logra un paso de avance en la integración de los procesos propios de la GEF y el Aseguramiento de la Universidad de La Habana y entra los resultados más significativos se pueden listar los siguientes:

- Enriquecimiento del lote de aplicaciones informáticos de la GEF en la UH.
- Integración paulatina con el sistema ASSETS usado en la GEF y el Aseguramiento de la UH.
- Documentación de los sistemas informáticos desarrollados.
- En cada uno de los sistemas informáticos desarrollados se realizó el Estudio de Factibilidad y de Costos mediante la metodología de Análisis por puntos de Casos de Uso.

CONCLUSIONES

Como resultado del presente trabajo se plantean las siguientes conclusiones:

1. Se realizó un estudio del estado del arte de la Gestión Económica Financiera en la Universidad de La Habana y de los temas de Análisis y Diseño de Sistemas Informáticos con énfasis en la Seguridad, haciéndose el correspondiente análisis bibliográfico.
2. Se desarrolló el análisis, diseño e implementación de las aplicaciones.
3. Se diseñaron y modelaron las bases de datos de cada una de las aplicaciones.
4. Se desarrollaron acciones de capacitación y entrenamientos en puestos de trabajo con cada una de las aplicaciones desarrolladas.
5. Existen otros sistemas en fase de desarrollo.
6. Se constató el aporte práctico de los sistemas presentados a partir del criterio de la Dirección de la Gestión Económica Financiera y el Aseguramiento de la UH.

REFERENCIAS

1. Autores, C. d. (2005). *Perfeccionamiento de la Actividad Económica. Sistemas y Subsistemas del Modelo de Gestión Económica financiera del Ministerio de Educación Superior*. La Habana (Cuba). Félix Varela.
2. MES, Dirección de informatización (2011). *Selección de documentos para la capacitación de profesores e investigadores en el manejo de la información electrónica*. Recuperado el 23/07/2012, de <http://169.158.24.166/tutoriales/Estrategia.html>
3. UH, VRE. (Noviembre de 2009). *Estrategia de Informatización para la Gestión Económica Financiera*. Recuperado el 23/07/2012, de <http://intranet.uh.cu/economica-y-financiera>

BIBLIOGRAFÍA CONSULTADA

Álvarez, Rubén. *Qué es SQL: Qué es y para qué sirve el SQL*. Recuperado el 05/05/2012, de <http://www.desarrolloweb.com/articulos/262.php?manual=9>

BIOCOM *Generalidades de Motores de bases de datos SQL Access, Oracle Sysbase, Mysql: Arquitectura Cliente-Servidor*. Recuperado el 23/07/2012, de http://www.biocom.com/sistema/base_de_datos/generalidades.html

González Urmachea, Mabel. *SQL Server*. <http://www.monografias.com/trabajos14/sqlserver/sqlserver.shtml> (4/4/2012)

Jacobson, Ivar; Booch, Grady; y, Rumbaugh, James. *El proceso unificado de desarrollo de software*. Editorial Félix Varela. La Habana, 2004. Tomos I, II.

MySQL AB. *Manual de referencia de MySQL*. Recuperado el 20/08/2012, de <http://www.mysql.com>

Pressman, Roger; *Ingeniería de software. Un enfoque práctico*. 2002. McGraw. Hill/Interamericana de España. Tomos I, II.

Stig Sæther Bakken; Alexander Aulbach; Egon Schmid; Jim Winstead; Lars Torben Wilson; Rasmus Lerdorf; Zeev Suraski; Andrei Zmievski; Jouni Ahto. *Manual de PHP*. Editado por el Grupo de documentación de PHP, 2002.

Vega, Aimée. *Aprenda Web Dinámico*. Instituto Cubano del Libro, Editorial Científico-Técnica. Ciudad de La Habana, Cuba, 2003.

CONSEJO EDITORIAL

COMPONENTES	
Director	Javier Francés Vilaplana
Editores adjuntos	Víctor Gisbert Soler María J. Vilaplana Aparicio Isabel Castillo Olmedo Vicente Sanchís Rico
Editor asociado	David Juárez Varón

COMITÉ CIENTÍFICO TÉCNICO

ÁREA TEXTIL	Prof. Dr. Josep Valldeperas Morell Universidad Politécnica de Cataluña
ÁREA FINANCIERA	Prof. Dr. Juan Ángel Lafuente Luengo Universidad Jaume I, Castellón de la Plana
ORGANIZACIÓN DE EMPRESAS Y RRHH	Prof. Dr. Francisco Llopis Vañó Universidad de Alicante
ESTADÍSTICA, INVESTIGACIÓN OPERATIVA	Prof. Dra. Elena Pérez Bernabéu Universidad Politécnica de Valencia
DERECHO	Prof. Dra. María del Carmen Pastor Sempere Universidad de Alicante
INGENIERÍA Y TECNOLOGÍA	Prof. Dr. David Juárez Varón Universidad Politécnica de Valencia
TECNOLOGÍA DE LA INFORMACIÓN Y LA COMUNICACIÓN	Prof. Dr. Manuel Llorca Alcón Universidad Politécnica de Valencia

