Integrated Support System for Information Management DATEC

Alexander Delgado Gutierrez, Ing.[1], Niurka Martínez Durán, Ing.[1], Jorge Luis Cuellar Mondeja, Ing.[1], Jorge Emilio Escala Maceo, Lic.[1]
[1] Universidad de las Ciencias Informáticas, Cuba, adgutierrez@uci.cu, nduran@uci.cu, jlcuellar2014@gmail.com, jeem@uci.cu

Abstract*– Systems for information management is a set of interrelated in order to pay attention to the control of information in organizations elements. This research aims to develop an integrated support system that allows help in managing the various flows of data handled by the center DATEC, Cuba. For this, a theoretical analysis of the subject matter took place, given the right tools and technologies for development. Functionalities that must meet the system were identified, the design is made from the selected patterns and the solution was implemented. A system that allows the information management areas of Fixed Assets, Information Security and Technology was obtained as a result. It makes it possible to also control access permissions and user roles dynamically; providing durability by possible changes in roles originally defined. Through trace records navigation around the authenticated personnel are obtained. The characteristics of the solution allows adapting to any environment with a similar business. Using black box testing, stress and load operation it was validated. The proposed solution is of great significance to the center and giving their managers a useful tool for decision making.

***Keywords*— information management, information management systems.**

Digital Object Identifier (DOI): http://dx.doi.org/10.18687/LACCEI2015.1.1.191
ISBN: 13 978-0-9822896-8-6
ISSN: 2414-6668

Sistema Integrado de Soporte para la Gestión de Información de DATEC

Ing. Alexander Delgado Gutierrez[1], Ing. Niurka Martínez Durán[2], Ing. Jorge Luis Cuellar Mondeja[3], Lic. Jorge Emilio Escala Maceo[4]
[1]Universidad de las Ciencias Informáticas, Cuba, adgutierrez@uci.cu, jlcuellar2014@gmail.com
[2] Universidad de las Ciencias Informáticas, Cuba, nduran@uci.cu, jeem@uci.cu

Resumen– Los sistemas para la gestión de información son un conjunto de elementos interrelacionados con el propósito de prestar atención al control de la información en las organizaciones. La presente investigación tiene como objetivo desarrollar un Sistema Integrado de Soporte que permita ayudar en la administración de los diversos flujos de datos manejados por el centro DATEC, Cuba. Para ello se realizó un análisis teórico del tema en cuestión, teniendo en cuenta las tecnologías y herramientas adecuadas para su desarrollo. Se identificaron las funcionalidades que debe cumplir el sistema, se realizó el diseño a partir de los patrones seleccionados y se implementó la solución. Se obtuvo como resultado un sistema que permite la gestión de la información de las áreas Activos Fijos, Seguridad Informática, y Tecnología. Hace posible también controlar el acceso, los permisos y los roles de los usuarios dinámicamente; lo que proporciona perdurabilidad ante posibles cambios en roles definidos inicialmente. A través de las trazas se obtienen los registros de la navegación de todo el personal autenticado. Las características propias de la solución permiten adaptarla a cualquier entorno con un negocio similar. Mediante pruebas de caja negra, stress y carga se validó su funcionamiento. La solución propuesta es de gran significación para el centro ya que confiere a sus directivos de una útil herramienta para la toma de decisiones.

Palabras claves — gestión de información, sistemas de gestión de información.

I. Introduction

La innovación y uso de las Tecnologías de la Información y las Comunicaciones (TIC) ha sido un proceso acelerado en el desarrollo científico técnico a escala mundial. De ellos depende en gran medida el desarrollo de la economía de los países en la actualidad. Se puede decir que las TIC no son más que aquel conjunto de tecnologías que permiten transmitir, procesar y difundir información de manera constante. Además constituyen las bases para sobre la cual se construye una sociedad de información y economía del conocimiento.

La Universidad de las Ciencias Informáticas (UCI) en Cuba vincula el proceso docente-educativo con la producción de sistemas informáticos. Además de lograr un continuo avance como institución académica y de investigación, posibilita una mejora constante en los sectores de la economía y la sociedad. Esto ha provocado una acertada transformación cuyas ventajas expresan el ahorro de recursos, la comunicación y la actualización de la información. La UCI presenta una estructura organizativa fragmentada en áreas por objetivos, una de ellas es la encargada de la producción de software, que agrupa a varios centros productivos entre los que se encuentra el Centro de Tecnologías de Gestión de Datos (DATEC).

El centro DATEC es el encargado de proveer soluciones integrales, productos y servicios relacionados con las tecnologías de gestión de datos, permitiendo la formación y capacitación de profesionales integrales con un alto nivel científico- productivo y el desarrollo de investigaciones afines a él. Dentro de su estructura tiene asociado el Grupo de Soporte el cual está conformado por tres áreas fundamentales: Tecnología, Seguridad Informática y Control de Activos Fijos.

El área de Tecnología se ocupa de gestionar los recursos informáticos de la entidad, sus prestaciones, cómo están distribuidos por locales y por puesto de trabajo. El área de Seguridad Informática verifica que estén implementadas todas las normas de seguridad regidas por el centro y la universidad. El área de Activos Fijos controla además de los recursos informáticos, los medios básicos asociados a los locales de la entidad. La presente investigación tiene como objetivo desarrollar el Sistema Integrado de Soporte para obtener una gestión de información general de las áreas descritas anteriormente para favorecer una toma de decisiones precisa acorde a las necesitades puntuales que se solicitan.

II. Materiales y Métodos

El valor de la presente investigación está dado en la posibilidad que ofrece a los directivos del centro DATEC de gestionar mediante el sistema informático que propone, todas las áreas del Grupo de soporte y visualizar mediante reportes información específica para la toma de decisiones que a tales efectos se aplica. Por otro lado el sistema por su arquitectura extensible se puede adaptar a cualquier entorno que posea un negocio similar.

Para darle solución al objetivo de la investigación se utilizó una combinación de métodos de nivel teóricos y empíricos:

Métodos de nivel teórico.

- Análisis y síntesis: Para analizar la información en la elaboración de los fundamentos teóricos en los que se sustenta la investigación y extraer los requerimientos del sistema.
- Histórico lógico: Empleado para el estudio sobre los Sistemas de Gestión de Información. Permitió expresar de forma teórica la importancia y necesidad de generar datos de prueba.

- Modelación: Utilizado durante toda la etapa de creación del software para la búsqueda de reglas o algoritmos y en la implementación de la herramienta.

Métodos de nivel empírico

- Revisión de documentos: Se empleó en lo fundamental para comprender y obtener la mayor cantidad de información con el fin de caracterizar las áreas Tecnología, Seguridad Informática y Activos fijos del Grupo de Soporte.
- Entrevista: Se utilizó para obtener la información necesaria referente al tema de investigación. Teniendo en cuenta al usuario final se escucharon las opiniones de expertos sobre cómo debe funcionar el Sistema Integrado de soporte, con el objetivo de recopilar elementos para el análisis del sistema.

Sistemas de gestión de información (SGI)

La gestión de la información implica determinar la información que se precisa; recoger y analizar la información; registrarla y recuperarla cuando sea necesaria; utilizarla; y divulgarla [1]. Un SGI es una herramienta mediante la cual los datos (entrada) se registran, almacenan, recuperan y procesan para la toma de decisiones (salida) [2].

Un SGI permite crear informes que proporcionan una visión completa de toda la información necesaria para tomar decisiones que van desde pequeños detalles diarios a nivel superior de estrategia. Los sistemas actuales de gestión de la información se basan en gran medida en la tecnología para recopilar y presentar datos, pero el concepto es más antiguo que las tecnologías informáticas modernas [3].

Metodología, Herramientas y Técnicas utilizadas

A continuación se presentan las descripciones de la metodología, herramientas y tecnologías utilizadas para dar solución al problema. Para definir la selección se tomaron en cuenta las necesidades existentes, tendencias actuales y el entorno donde se desplegará el Sistema.

Las metodologías de desarrollo de software son un conjunto de procedimientos, técnicas, herramientas y soporte documental para el desarrollo de un producto (software). Se basan en una combinación de los modelos de proceso genéricos (cascada, evolutivo e incremental). Son para estructurar, planear y controlar el proceso de desarrollo del software. Constituyen una guía que define las tareas y actividades que se deben realizar para obtener un software de buena calidad [4].

Las metodologías ágiles son orientadas a la interactuación con el cliente y el desarrollo incremental del software. Mostrando versiones parcialmente funcionales de la aplicación al cliente en intervalos cortos de tiempo, para que pueda evaluar y sugerir cambios en el producto según se va desarrollando. Se enmarca más en la capacidad de respuesta ante un cambio realizado que en el seguimiento estricto de un plan. Consiste básicamente en ajustarse estrictamente a una serie de reglas que se centran en las necesidades del cliente para lograr un producto de buena calidad en poco tiempo. Entre ellas se destacan: XP (eXtreme Programming), Scrum, ASD (Adaptive Software Development) y OpenUP.

Se decide seleccionar a OpenUP para enfrentar el desarrollo del software propuesto teniendo en cuenta que permite detectar errores con prontitud a través de un ciclo iterativo y tiene un enfoque orientado al cliente con iteraciones cortas. El proceso es mínimo en que solamente el contenido fundamental es incluido; es completo en que puede ser manifestado como todo el proceso para construir un sistema; extensible en que puede ser utilizado como fundamento sobre el cual el contenido de proceso se pueda agregar o adaptar según lo necesitado. OpenUP como metodología de desarrollo es conducida por el principio de colaboración para alinear intereses y para compartir su comprensión [5]. También es apropiada para proyectos pequeños y de bajos recursos permite disminuir las probabilidades de fracaso en los proyectos pequeños e incrementar las probabilidades de éxito.

Teniendo en cuenta esto, se exhibe una síntesis de las tecnologías, herramientas a utilizar durante la elaboración del software fundamentando su selección. Se decidió usar como lenguaje de programación del lado del cliente JavaScript auxiliándose de la librería de ExtJS 3.4.0 y el lenguaje de programación PHP del lado del servidor. Se estableció utilizar como framework de desarrollo Symfony2.3, como IDE el NetBeans 7.4, como gestor de base de datos PostgreSQL 9.2, como herramienta CASE Visual Paradigm 6.4. Se definió como herramienta de prueba Jmeter 2.3.4 y como generador de reportes JasperReports 4.5. Este ambiente de desarrollo seleccionado apoya la implementación del sistema propuesto, brindando agilidad y variedad de funciones a los desarrolladores.

III. Resultados y Discusión

Sistema Integrado de Soporte. Elementos técnicos

El término requisitos de software puede conceptualizarse acorde a lo planteado en la traducción certificada de la ISO 9000: 2000: se afirma que los clientes necesitan productos con características que satisfagan sus necesidades y sus expectativas y se expresan en la especificación del producto y son generalmente denominadas como requisitos del cliente. Los requisitos son la necesidad o expectativa establecida, generalmente implícita u obligatoria [6].

Los requisitos funcionales definen el comportamiento interno de un software, son condiciones que el sistema ha de cumplir. Estos muestran las funcionalidades que deben satisfacerse para cumplir con las especificaciones de software [7]. Para el desarrollo del Sistema Integrado de Soporte del

Centro DATEC se definieron 48 requisitos funcionales, lo que permitió una orientación para la creación del producto.

Los requisitos no funcionales son restricciones en el servicio o función ofrecida por el Sistema [8]. Incluyen restricciones de tiempo, del proceso de desarrollo y estándares. Especifican propiedades del sistema, como confiablidad y seguridad [9].

Ejemplos de requisitos no funcionales identificados:

Software.

El servidor donde se instalará la aplicación debe cumplir con los siguientes requisitos: Sistema Operativo:

- GNU/Linux preferentemente Ubuntu GNU/Linux 1 2.04 o superior, Debian 6 GNU/Linux o superior.
- Paquetes: apache2, php5, libapache2-mod-php5, php5-cli, php5-pgsql, php5-sqlite, libssh2-php, libssh2-1 -dev, php5-soap, php-apc, php5-curl, ssh.
- Navegador: Mozilla Firefox versión 4.0 o superior.

El servidor donde se instalará la base de datos debe cumplir con los siguientes requisitos:

- Sistema Operativo: GNU/Linux preferentemente Ubuntu GNU/Linux 1 2.04 o superior, Debian 6 GNU/Linux o superior.
- PostgreSQL versión 9.0 o superior.
- PgAdmin III o algún administrador para PostgreSQL.
- PostgreSQL debe estar correctamente configurado para aceptar conexiones vía TCP/IP utilizando el método de autenticación por md5.

Hardware

Las PC clientes debe cumplir con los siguientes requisitos de hardware:

- Ordenador Pentium IV o superior, con 1.7 GHz de velocidad de microprocesador.
- Memoria RAM mínimo 256 MB.
- Un mínimo de 1GB de espacio en disco.

Las PC Servidor debe cumplir con los siguientes requisitos de hardware:

- Ordenador Pentium IV o superior, con 1.7 GHz de velocidad de microprocesador. Memoria RAM mínimo 1Gb.
- Disco Duro con 10Gb de capacidad para instalar el sistema.

Eficiencia

El sistema responderá de manera inmediata siempre que se cumplan los requisitos de hardware necesarios:

- Para 100 peticiones concurrentes el sistema responderá en menos de 3 segundos.
- Para 250 peticiones concurrentes el sistema responderá en menos de 8 segundos.

Portabilidad

- El sistema deberá ser desarrollado de forma tal que sea multiplataforma.

Seguridad.

- La información solo podrá ser vista por los usuarios con el nivel de acceso requerido para ello; permitiendo que las funcionalidades del sistema se muestren de acuerdo a lo permisos del usuario que esté activo.

El diagrama de casos de uso documenta el comportamiento de un sistema desde el punto de vista del usuario. Por tanto los casos de usos determinan los requisitos funcionales del sistema, los cuales representan las funcionalidades que un sistema puede ejecutar [10].

Un caso de uso es una secuencia de interacciones que se desarrollarán entre un sistema y sus actores en respuesta a un evento que inicia un actor principal sobre el propio sistema. Los diagramas de casos de uso sirven para especificar la comunicación y el comportamiento de un sistema mediante su interacción con los usuarios u otros sistemas [10].

Los actores pueden representar roles jugados por usuarios humanos, hardware externo, u otros sujetos. Un actor no necesariamente representa una entidad física específica, sino simplemente una faceta particular de alguna actividad que es relevante a la especificación de sus casos de uso asociados.

TABLA I
DESCRIPCIÓN DE LOS ACTORES DEL SISTEMA

Actor	Descripción
Administrador	Es el encargado de administrar todos los usuarios existentes en el sistema y de otorgarle los roles a los mismos. Además tiene acceso a todas las funcionalidades del sistema.
Usuario	Son todos los usuarios del sistema. Sus funcionalidades en el mismo están determinadas por el rol que le conceda el administrador.
LDAP	El Protocolo Ligero de Acceso a Directorios (Lightweight Directory Access Protocol, LDAP por sus siglas en inglés) es un conjunto de protocolos de acceso a directorios de información. De él se utilizan las prestaciones para la autenticación, regidas por los elementos: usuario y contraseña, que representan a todos los usuarios del dominio UCI.

Se puede decir que un diagrama de casos de usos documenta el comportamiento de un sistema desde el punto de vista del usuario. Muestra cómo los actores del sistema se relacionan con las funcionalidades desde su entrada en el mismo.

En la siguiente figura se muestran 10 casos de uso del sistema, de ellos 4 son casos de usos significativos (Gestionar Recurso del local, Gestionar Prestación, Gestionar Usuario, Gestionar Rol). El actor Administrador tendrá la tarea de asignar los roles y los permisos a los usuarios que intervienen en el proceso a través del caso de uso Gestionar Rol. Este es el único que puede realizar todas las funcionalidades del sistema. El actor Usuario realizará las funcionalidades definidas en los permisos del rol que represente.

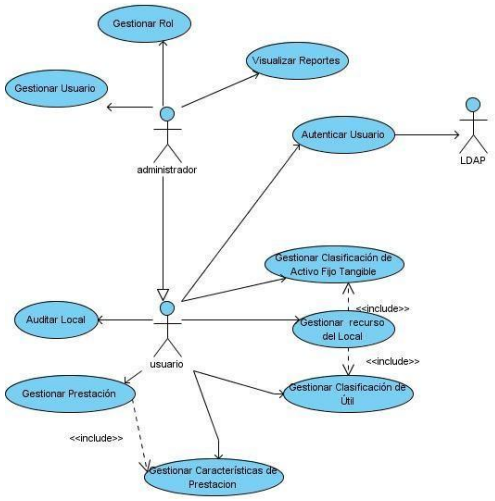

Fig. 1 Diagrama de Casos de Usos del Sistema Integrado Se Soporte.

Se definieron las relaciones entre las funcionalidades a través de relaciones de inclusión (<<include>>). Se utilizaron los patrones CRUD (Crear, Mostrar, Modificar y Eliminar) que se aplican cuando se quiere realizer altas, bajas, cambios y consultas a algunas de las entidades del sistema e inclusión concreta que se aplica cuando un flujo puede extender del flujo de otro caso de uso así como ser realizado en sí mismo. Se tuvo como relación entre los actores el patrón múltiples actores, donde el actor usuario puede tener el mismo rol sobre los casos de usos del administrador en dependencia de sus permisos. Se utilizó también el patron múltiples actores roles diferentes ya que el caso de uso Autenticar Usuario interactúa con actor Usuario y con LDAP.

Arquitectura y Patrones

Para la creación del sistema se utilizó el framework Symfony el cual propone una arquitectura Modelo-VistaControlador (MVC), para una mejor organización y manejo de las vistas. Este patrón separa los datos de una aplicación, la interfaz de usuario y la lógica de control en tres componentes distintos.

- Modelo: Agrupa los datos y la lógica de la aplicación.
- Vista: Se encarga de mostrar los datos recogidos en el modelo, generalmente a través de una interfaz.
- Controlador: Se encarga de procesar las interacciones con el usuario y realizar los cambios apropiados en el modelo o en la vista.

Ventajas: Brinda soporte de vistas múltiples, es decir, que la interfaz de usuario puede mostrar multiples vistas de los mismos datos simultáneamente, debido a que la vista se encuentra separada del modelo y no existe dependencia directa entre ellos. En el sistema las clases JS generan las interfaces formando parte de la capa Vista. Las clases Controller forman la capa Controladora, debido a que son estas las encargadas de responder las peticiones de los usuarios. Por último la capa Modelo la integrarán las clases EntityRepository, teniendo en cuenta que ellas representan los datos del dominio de la aplicación.

GRASP es un acrónimo que significa General Responsibility Asignment Software Patterns (patrones generales de software para asignar responsabilidades). Estos describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones.

Experto: Asigna una responsabilidad a un experto en información: la clase que cuenta con la información necesaria para cumplir la responsabilidad. Experto es un patrón que se usa más que cualquier otro al asignar responsabilidades; es un principio básico que suele utilizarse en el diseño orientado a objetos. Con él no se pretende designar una idea oscura ni extraña; expresa simplemente la "intuición" de que los objetos hacen cosas relacionadas con la información que poseen [11]. Este patrón se aplica en la clase controladora AdicionarRecurso del Sistema, la cual es experta en la realización de su funcionalidad que consiste en adicionar un nuevo recurso al sistema.

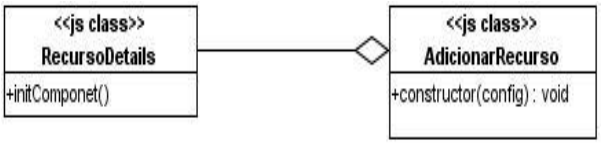

Fig. 2 Patrón Experto.

Creador: El patrón creador guía la asignación de responsabilidades relacionadas con la creación de objetos, tarea muy frecuente en los sistemas orientados a objetos. El propósito fundamental de este patrón es encontrar un creador que debemos conectar con el objeto producido en cualquier evento. Al escogerlo como creador, se da soporte al bajo acoplamiento [11]. Este patrón se utiliza en la clase ManagePlugin.js, clase en JavaScript que permite agregar comportamientos específicos a la clase EliminarUsuario.js que interactúa con ella.

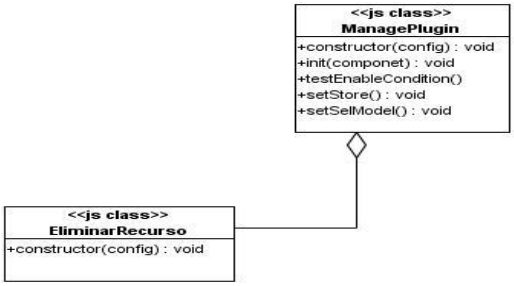

Fig. 3 Patrón Creador.

Bajo Acoplamiento: Es un patrón evaluativo que el diseñador aplica al juzgar sus decisiones de diseño. Soporta el diseño de clases más independientes, que reducen el impacto de los cambios, y también más reutilizables, que acrecientan la oportunidad de una mayor productividad [11]. Se aplica en las clases del modelo, pues existe poca dependencia entra las clases de acceso a datos y las de abstracción de datos, lo que permite mayor reutilización. Se pueden modificar las clases del modelo sin que se afecten las del controlador.

Alta Cohesión: Propone asignar responsabilidades a las clases evitando que estas realicen un trabajo excesivo y que las tareas no sean afines. Es un patrón evaluativo que el desarrollador aplica al valorar sus decisiones de diseño [11]. Por ejemplo este patrón se aplica en la clase EditarRecurso debido a que es una de las encargadas de mostrar a la clase RecursoDetails la información del recurso seleccionado.

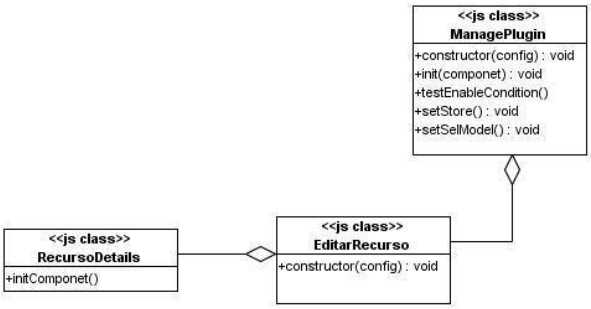

Fig. 4 Patrón Alta Cohesión.

Controlador: El patrón controlador resuelve elproblema: ¿Quién debería ser el responsable de gestionar un evento de entrada al sistema?. Este asigna una responsabilidad de recibir o manejar un mensaje de evento del sistema a una clase. Posibilita la organización y claridad en el diseño. Favorece el bajo acoplamiento [11]. Este patrón se ve aplicado en todas las clases controladoras ya que son las encargadas de ejecutar eventos de entradas externas en el sistema (Adicionar, Modificar, Listar, Eliminar y Mover).

Patrones GOF

Fachada: Proporciona una interfaz unificada para un conjunto de interfaces de un subsistema. Define la interfaz de alto nivel que hace que sea más fácil de utilizar el subsistema. Se utiliza en la clase AppSIS la cual hace función de interfaz principal de la aplicación y a la vez es la responsable de crear las interfaces visuales de los módulos Seguridad y Centro, proporcionando una interfaz de alto nivel que hace que la aplicación sea más fácil de usar.

Decorador: Añade dinámicamente nuevas responsabilidades a un objeto, proporcionando una alternativa flexible a la herencia para extender funcionalidades. Este patrón se evidencia en la clase FormRecurso porque este podrá ser modificado en dependencia del tipo de recurso a adicionar.

Singleton: Garantiza que una clase sólo tenga una instancia, y proporciona un punto de acceso global a ella. Este patrón se utiliza en la clase Centro donde se crea una única instancia de cada clase controladora en la vista.

Observador: Define una dependencia de uno a muchos entre objetos, de forma que cuando un objeto cambia de estado se notifica y actualizan todos los objetos automáticamente. Se aplica en la clase ManagePlugin ya que esta se mantiene observando a la clase Recursos en espera de que esta notifique que un recurso ha sido seleccionado para activar las funcionalidades de editar y eliminar.

Adaptador: Convierte la interfaz de una clase en otra distinta que es la que esperan los clientes. Además permiten que cooperen clases que de otra manera no podrían hacerlo, pues tendrían interfaces incompatibles.

Constructor: Separa la construcción de un objeto complejo de su representación, de forma que el mismo proceso de construcción pueda crear diferentes representaciones [7].

Funcionalidades del sistema

La siguiente figura muestra la interfaz gráfica correspondiente a caso de uso Autenticar Usuarios, utilizada para los usuarios que deseen acceder a los servicios del Sistema Integrado de Soporte del Centro DATEC. Posee tres campos no opcionales para la entrada de datos: usuario, contraseña y modo de autenticación. Para esto se definió un proveedor de autenticación personalizado el cual recoge los datos capturados del formulario. Si la autenticación del usuario es de manera local la validación se realiza en dependencia del usuario registrado en el sistema, de otra manera si el usuario se autentica por el dominio los datos se validan en el servidor del dominio accediendo al servicio LDAP de la universidad.

Fig. 5 Autenticar Usuario.

En el Sistema Integrado de Soporte del Centro DATEC, a los recursos de Activo Fijo Tangible CPU se les realizan

auditorías de seguridad informática y se le actualizan dinámicamente las prestaciones, donde el usuario selecciona los recursos que desea actualizar o auditar, e introduce los datos de acceso enviándolos al servidor [Ver figura 6]. El servidor establece una conexión con las computadoras por el protocolo ssh y obtiene la información necesaria a través de la ejecución de un script en bash, dependiendo en gran medida del sistema operativo instalado. Durante el trascurso de la operación se muestra el progreso [Ver figura 7]. La implementación de estas funcionalidades permite al usuario un consumo no excesivo de sus horas de trabajo proporcionándole tiempo para el desarrollo de otras actividades.

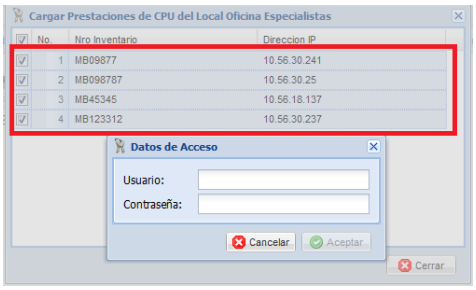

Fig. 6 Cargar prestaciones de CPU.

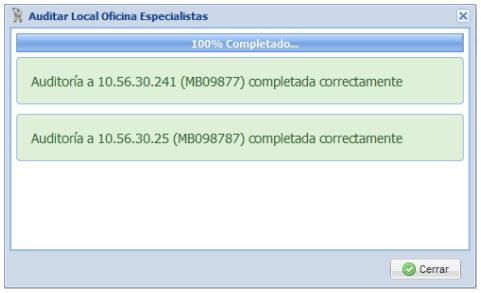

Fig. 7 Auditar Local.

Pruebas de Software

Las pruebas no pueden asegurar la ausencia de errores; sólo puede demostrar que existen defectos en el software [12]. Entre las pruebas que se le realizan al software están las pruebas de caja negra y la pruebas de rendimiento (carga y Stress).

Pruebas de Caja Negra

Las pruebas de caja negra denominadas pruebas de comportamientos, se centran en los requisitos funcionales del software. Estas permiten derivar conjuntos de condiciones de entrada que ejercitarán por completo todos los requisitos funcionales. Las pruebas de caja negra tratan de encontrar errores en las siguientes categorías:

- Funciones incorrectas o faltantes.
- Errores de Interfaz.
- Errores de estructura de datos o base de datos externas.
- Errores de comportamiento y desempeño.
- Errores de inicialización y término.

Existen diversos métodos de caja negra en los cuales se encuentra los métodos gráficos de prueba, análisis de valores límite y partición equivalente. La partición equivalente es un método de prueba de caja negra que divide el dominio de la entrada de un programa en la clase de datos a partir de los cuales pueden derivarse casos de prueba. Se esfuerza por definir un caso de prueba que descubra ciertas clases de errores. Las clases de equivalencia se definen de acuerdo a las siguientes directrices:

- Si una condición de entrada especifica un rango, se definen una clase de equivalencia válida y dos no válidas.
- Si una condición de entrada requiere de un valor específico, se definen una clase de equivalencia válida y dos no válidas.
- Si una condición de entrada especifica un miembro de un conjunto, se define una clase de equivalencia válida y otra no válida.
- Si una condición de entrada es booleana, se define una clase de equivalencia válida y otra no válida [13].

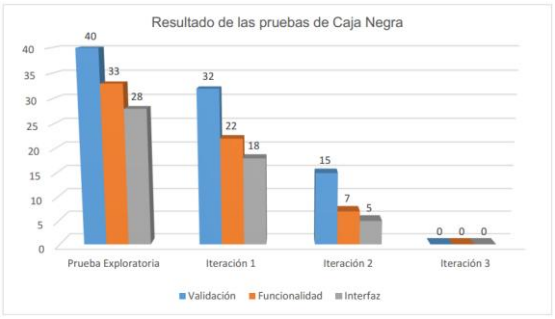

Fig. 8 Resultado de las Pruebas de Caja Negra.

En el gráfico anterior se muestra los principales datos obtenidos durante la validación del sistema, donde se aprecia una prueba exploratoria en la que se encontraron 40 errores de no validación, 33 de funcionalidad y 28 de interfaz. En la primera y segunda iteración se corrigieron estos errores y al realizarse la tercera iteración no se detectaron errores. Estas pruebas permitieron comprobar el correcto funcionamiento del módulo y la correcta validación de los campos, verificando que solo acepten los caracteres válidos.

Pruebas de rendimiento (Carga y Stress)

El objetivo de las pruebas de rendimiento es determinar si el usuario está satisfecho con la velocidad de la aplicación. Las pruebas de carga determinan el volumen de trabajo necesario para que el sistema funcione en hora punta y la de stress sirve para obtener datos, sobre la carga del sistema, que ayuden a realizar el dimensionamiento del Sistema [14]. En la tabla mostrada en la figura siguiente se puede observar que los

tiempos de respuestas obtenidos son expresados en milisegundos.

Las pruebas de rendimiento se realizaron con la herramienta de pruebas Jmeter 2.3.4. Para la ejecución de la prueba de carga se tomó una muestra de 100 personas obteniendo como resultado un tiempo que no sobrepasa los 3 segundos. Además para la ejecución de la prueba de stress se tomó una muestra de 250 obteniéndose como resultado un tiempo que no sobrepasa los 8 segundos.

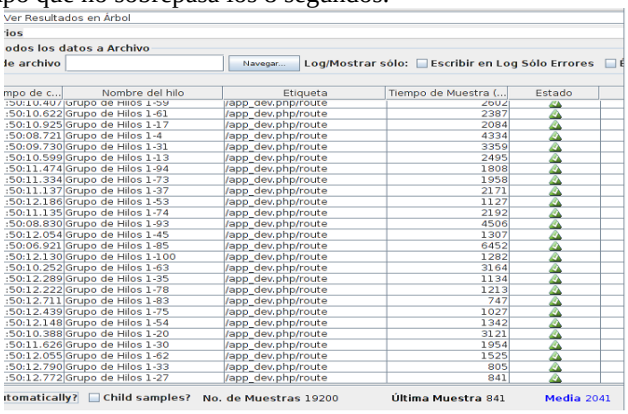

Fig. 9 Resultado de las Pruebas de Carga.

El diagrama de despliegue muestra las relaciones físicas entre los componentes hardware y software en el sistema final. Es un grafo de nodos unidos por conexiones de comunicación donde cada nodo puede contener instancias de componentes. En general un nodo puede ser una unidad de computación de algún tipo, desde un sensor hasta una computadora central y las instancias de componentes de software pueden estar unidas por relaciones de dependencia [15].

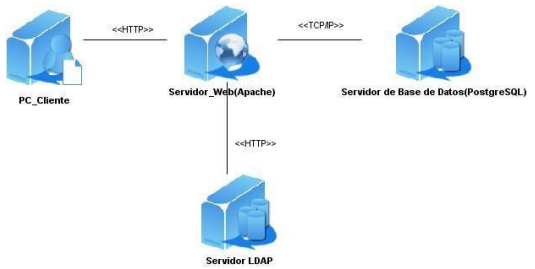

Fig. 10 Modelo de despliegue.

Descripción de los nodos

- Nodo PC_Cliente: Ordenador desde donde el cliente accede al sistema.
- Nodo Servidor Web: Ordenador donde se ejecuta el sistema.
- Nodo Servidor de Base Datos: Almacena la información que se obtiene del sistema.
- Nodo Servidor LDAP: Base de datos donde se encuentran todos los usuarios de la universidad.
- HTTP: Protocolo de transferencia utilizado para conectar la computadora del cliente con el servidor donde está el sistema y este último con el servidor de los usuarios de la universidad (LDAP).
- TCP/IP: Protocolo para conectar el servidor de aplicaciones con las bases de datos.

De manera general se puede decir que el Sistema Integrado de Soporte constituye una solución informática integral siguiendo el enfoque de arquitectura empresarial. Integra las cuatro capas fundamentales para el desarrollo de sistemas de información como son: el negocio, los datos, las aplicaciones y la tecnología [16].

Arquitectura de Negocio: Define las estrategias y procesos claves del negocio. DATEC crea servicios informáticos que agrupa tanto a los sistemas de información, como a los denominados sistemas de inteligencia empresarial o de negocios, cuyo propósito fundamental es apoyar el proceso de toma de decisiones de las diferentes entidades. Esto requiere fundamentalmente que la distribución de los recursos de cómputo sea acorde a las prioridades que se presenten para satisfacer de manera gradual las demandas realizadas al centro. Esta aplicación permite la gestión de departamentos, proyectos, locales, recursos. También posibilita la gestión de los usuarios y sus categorías, puestos de trabajo y usuarios por puesto de trabajo permitiendo realizar una gestión más amplia sobre los recursos ubicados en los locales y proyectos del usuario. Para una mejor seguridad los usuarios del sistema también tienen la opción de cambiar sus contraseñas y se emplea la política de control de acceso basado en roles. La visualización de trazas y de reportes son también actividades contempladas en él. Además posibilita la realización de las auditorías informáticas. Este sistema posee además otras funcionalidades como la gestión de incidencias por puesto de trabajo.

Arquitectura de Datos: Específica cómo administrar los datos del negocio. El sistema propuesto permite que la actualización de las prestaciones que brindan los recursos de cómputo pueda ser actualizada tanto manualmente como por el usuario. Los roles asignados a los usuarios pueden ser actualizados. Se generan y actualizan expedientes técnicos, lo que constituye un control preciso de cada recurso de cómputo en la organización. Este sistema posee un Control de Activos Fijos que favorece el control total de todos los recursos computacionales o no. Todo esto es posible mediante su robusto diseño de base de datos integrado al diseño arquitectónico Modelo Vista Controlador.

Arquitectura de Aplicaciones: Especifica un diagrama para cada sistema de aplicación detallando que las interacciones entre ellos y los procesos del negocio. El sistema propuesto consta de un conjunto de aplicaciones que soportan sus funcionalidades. Presenta servicios de intercambio de datos gráficos, así como funciones de procesamiento de texto y documentos. Es un sistema extensible al cual se le pueden agregar varios módulos.

Arquitectura Tecnología (TI): Describe los componentes de Hardware, software, comunicaciones y de redes necesarios para soportar el núcleo del negocio. El sistema propuesto presenta una buena arquitectura tecnológica, descrita anteriormente, la misma está alineada con estándares internacionales utilizando tecnología actualizada.

El sistema mejora las relaciones de los departamentos, actores involucrados en pro de unificar criterios para alcanzar los objetivos generales del negocio. Identifica oportunidades en cada uno de los proyectos de DATEC, mediante los diferentes análisis y puntos de vista.

IV. CONCLUSIONES

Con la realización de esta investigación se le dio cumplimiento al objetivo previamente planteado:

- El estudio de los sistemas existentes en el mundo vinculados con la gestión de información permitió comprender la estructura de estos sistemas.
- El estudio y selección de la metodología, herramientas y tecnologías permitió sentar las bases para el desarrollo del Sistema Integrado de Soporte del Centro DATEC.
- La implementación del Sistema Integrado de Soporte del Centro DATEC dio cumplimiento a los 48 requisitos funcionales definidos y los no funcionales identificados.
- El diseño y puesta en práctica de las pruebas de software permitió comprobar el funcionamiento del Sistema Integrado de Soporte del Centro DATEC.
- Se obtuvo un Sistema Integrado de Soporte del Centro DATEC que centralizó las áreas de Grupode Soporte, el cual por su arquitectura extensible se puede adaptar a cualquier entorno que posea un negocio similar.

REFERENCES

[1] AGUILAR F., S. 1997. El reto del medio ambiente: conflictos e intereses en la política medioambiental europea.,Madrid, Alianza Editorial, D.L

[2] KRISHNAN, A. N., BARIDALYNE, YADAV, K., SINGH, S. & GUPTA, V. 2010. Evaluation of computerized health management information system for primary health care in rural India. BMC Health Services Research, X.

[3] INGRAM, D. 2013. What Is a Management Information System? The Houston Chronicle

[4] Flores, Carmina Lizeth Torres. Establecimiento de una Metodología de Desarrollo de Software. 2008.

[5] Cockburn, Alistair. Agile Software Development. 2006

[6] ISO 9000:2000. ISO 9000:2000 Sistemas de gestión de la calidad-Fundamentos y vocabulario. 2000.

[7] Somerville, Ian. Ingeniería de Software 7ma Edicion. 2011.

[8] SEWBOK, Guide to the Software Engineering Body of Knowledge. 2004.

[9] YOUNG, R. The Requirements Engineering Handbook. 2004.

[10] Tello, Jesus Caceres. [En línea][Cita del 07 de 04 de 2014.] http://www2.uah.es/jcaceres/capsulas/DiagramaCasosDeUso.pdf.

[11] Larman, Graig. UML y Patrones. 2003.

[12] Pressman, Roger S. Ingeniería de Software. Un enfoque práctico. 6ta Edición. s.l.: McGraw-Hill Companies, 2007. ISBN: 8448132149.

[13] Pressman, Roger. Técnicas de Prueba. . s.l. : Vol. Cap 14, parte 1.

[14] Globe TESTIN. Globe TESTIN. [En línea] [Citado el: 29 de 04 de 2014.] http://www.globetesting.com/pruebas-de-rendimiento/ .

[15] Vilas, Ana Fernandez. Diagrama de Despliegue. Diagrama de Despliegue. [En línea] http://www-gris.det.uvigo.es/~avilas/UML/node50.html.

[16] The Open Group. The open group (Merking standards work). [En línea] [Citado el: 08 de 02 de 2011.] http://www.togaf.info/togaf9/togafSlides9/TOGAF-V9-M2-TOGAF9-Components.pdf.