

Tipo de artículo: Artículo original
Temática: Desarrollo de sistemas informáticos
Recibido: 22/09/2016 | Aceptado: 28/09/2016

Generador de rutinas para el Sistema Automatizado para la Gestión de la Información

Generator routines for the Automated System for Information Management

Omar Mar Cornelio, Miguel Angel Cárdenas Hernández, Lijandy Jiménez Armas

Dpto. de Programación, Facultad de Ciencias y Tecnologías Computacionales, Universidad de las Ciencias Informáticas. Cuba. omarmar@uci.cu

Centro de Tecnologías y Gestión de Datos, Facultad de Ciencias y Tecnologías Computacionales, Universidad de las Ciencias Informáticas. Cuba. macardenas@uci.cu

Centro de Tecnologías y Gestión de Datos, Facultad de Ciencias y Tecnologías Computacionales, Universidad de las Ciencias Informáticas. Cuba. ljimenez@uci.cu

Autor para correspondencia: omarmar@uci.cu

Resumen

La estadística adquiere relevancia en tanto hace posible transformar grandes cantidades de datos en información útil a los procesos administrativos como soporte a la toma de decisiones. En Cuba la Oficina Nacional de Estadística e Información (ONEI) realiza la gestión de los datos estadísticos a través del Sistema Automatizado para la Gestión de la Información (SAGI), desarrollado en convenio con la Universidad de las Ciencias Informáticas (UCI). Este sistema permite el diseño de formularios y encuestas para la captura de información y la generación de reportes mediante el Generador Dinámico de Reportes (GDR), lo que contribuye al proceso de toma de decisiones. Uno de los procesos fundamentales para la concepción de reportes es la generación de rutinas, algo que no es posible realizar hoy en SAGI. Este proceso se encuentra implementado en una herramienta externa, SIGETools, lo que obliga a los especialistas a utilizar dos o más aplicaciones para la obtención de reportes. El objetivo de la presente investigación es desarrollar un módulo para SAGI que permita la creación de rutinas. Para guiar dicha investigación se seleccionó como metodología de desarrollo de software OpenUP, UML 2.1 como lenguaje de modelado y Visual Paradigm 8.0 como herramienta CASE., los lenguajes de programación PHP 5 y JavaScript, Symfony 1.1.7 y ExtJS 3.4 como framework de desarrollo y para el tratamiento de las bases de datos PostgreSQL 9.3 y PgAdmin III. Además, se

diseñaron e implementaron las clases del módulo aplicando buenas prácticas de los patrones de diseño. Se le realizaron pruebas al mismo, comprobando su correcto funcionamiento.

Palabras clave: Datos estadísticos; rutinas; reportes; sistema automatizado para la gestión de información

Abstract

In Cuba's National Office of Statistics and Information (ONEI) make the management of statistical data through the Automated System for Information Management (SAGI), created in partnership with the University of Information Science (UCI). This system allows the design of survey templates and forms to capture statistical data, but does not allow the generation of routines, one of the fundamental processes for reporting. Today this process is implemented in an external tool, which brings with it a cumbersome job as specialists use two applications for creating reports. The aim of this research is to develop a module for SAGI that allows the creation of routines. To guide the development process module was selected as software development methodology OpenUP, as UML 2.1 Modeling Language 8.0 and Visual Paradigm as CASE tool. It was used for the implementation languages PHP 5 and JavaScript, ExtJS Symfony 1.1.7 and 3.4 programming and development framework and for the treatment of databases PostgreSQL 9.3 and PgAdmin III. In addition, they were designed and implemented classes applying good practices module design patterns. Tests were performed at the same, in order to check for proper operation.

Keywords: Statistical data, routines, reports, Automated System for Information Management.

Introducción

El análisis estadístico resulta fundamental para la recolección y agrupamiento de datos de diversos tipos para construir con ellos informes estadísticos que nos permitan conocer y analizar el comportamiento sobre diferentes temas, siempre desde un punto de vista cuantitativo. En Cuba dado el carácter esencialmente central de la administración del Estado, toda empresa, organismo, organización o institución que posea información de interés para la toma de decisiones de los órganos de la administración del Estado, está en la obligación de reportar información estadística bajo convenio con la Oficina Nacional de Estadística e Información (ONEI).

La ONEI ha priorizado la introducción de técnicas de almacenamiento y procesamiento de datos. En este contexto se inició de conjunto con la Universidad de las Ciencias Informáticas (UCI) el desarrollo de una aplicación quede

soporte al marco metodológico del trabajo estadístico. Surgiendo el proyecto de informatización Sistema Automatizado para la Gestión de la Información (SAGI), este tiene como finalidad la gestión estadística en Cuba. Como parte de esta gestión comprende los procesos de diseño de plantillas de encuestas y formularios, para la captura de datos estadísticos, así como también permite organizar y exhibir los datos mediante reportes. Estos permiten dar un formato determinado a la información captada para mostrarla por medio de un diseño atractivo y que sea fácil de interpretar por los usuarios.

SAGI genera los reportes mediante el Generador Dinámico de Reportes (GDR) en su versión 1.8, para la obtención de los mismos el especialista debe crear un modelo (abstracción de la base de **datos que contiene tablas, vistas y funciones**). También debe implementar las consultas necesarias para obtener los datos que se desean visualizar a través del reporte. Posteriormente se selecciona un modelo y se procede a actualizar los metadatos de este, de manera que se puedan incluir las consultas implementadas o modificadas. En ocasiones los reportes difieren en algún parámetro, dígame provincia, mes, año, indicador y nombre de la empresa, aun cuando la diferencia entre los reportes sea mínima es necesario la repetición de todo el proceso para cada uno de estos, ya que la ONEI procesa anualmente un gran cúmulo de datos lo que trae como consecuencia que se emplee mucho tiempo y personal dedicado a ello.

Actualmente la ONEI para dar solución a esta situación emplea SIGETools, una aplicación que facilita la generación de funciones parametrizadas o rutinas, las cuales son funciones SQL que permiten parámetros de diferentes tipos, posibilitando la creación de reportes y a su vez estas puedan ser reutilizadas por los especialistas de la entidad. Aun cuando SIGETools ha agilizado el trabajo con los reportes hasta el momento, esta presenta varios inconvenientes:

- Está desarrollada con un lenguaje de licencia privativa, lo cual no cumple con las políticas de soberanía tecnológica en Cuba.
- Es una herramienta externa al SAGI la cual hace el trabajo más engorroso ya que se utilizan dos herramientas totalmente diferentes con el mismo fin.
- Es una herramienta de escritorio, por lo que debe ser instalada en todos los ordenadores que se utiliza el sistema.
- No es compatible con todas las distribuciones de Microsoft Windows y ninguna de Linux.
- No existe documentación sobre la herramienta, lo que dificulta el trabajo con la misma.
- No cuenta con un ciclo de soporte.

El empleo de dos herramientas totalmente diferentes tecnológicamente trae consigo la necesidad de una cantidad más elevada de especialistas con mayor tiempo disponible y mayor capacitación.

Materiales y métodos

La metodología para el desarrollo de software en un modo sistemático de realizar, gestionar y administrar un proyecto para llevarlo a cabo con altas posibilidades de éxito. Una metodología para el desarrollo de software comprende los procesos a seguir sistemáticamente para idear, implementar y mantener un producto software desde que surge la necesidad del producto hasta que se cumple el objetivo por el cual fue creado (INTECO, 2010).

Una definición estándar de metodología puede ser el conjunto de métodos que se utilizan en una determinada actividad con el fin de formalizarla y optimizarla. Determina los pasos a seguir y cómo realizarlos para finalizar una tarea (INTECO, 2010), dentro de estas metodologías se encuentra OpenUP.

OpenUP: Es un proceso modelo y extensible, dirigido a gestión y desarrollo de proyectos de software basados en desarrollo iterativo, ágil e incremental apropiado para proyectos pequeños y de bajos recursos; y es aplicable a un conjunto amplio de plataformas y aplicaciones de desarrollo (eclipse, 2011).

Se escoge esta metodología ya que:

- Es apropiado para proyectos pequeños y de bajos recursos permite disminuir las probabilidades de fracaso en los proyectos pequeños e incrementar las probabilidades de éxito.
- Permite detectar errores tempranos a través de un ciclo iterativo.
- Evita la elaboración de documentación, diagramas e iteraciones innecesarios requeridos en la metodología RUP.
- Por ser una metodología ágil tiene un enfoque centrado al cliente y con iteraciones cortas.

Lenguaje de Programación:

Un lenguaje de programación es aquel elemento dentro de la informática que permite crear programas mediante un conjunto de instrucciones, operadores y reglas de sintaxis; que pone a disposición del programador para que este pueda comunicarse con los dispositivos de hardware y software existentes (Marin, 2008).

PHP5: Acrónimo de "PHP: Hypertext Preprocessor", es un lenguaje de 'scripting' de propósito general y de código abierto que está especialmente pensado para el desarrollo web y que puede ser embebido en páginas HTML. Su sintaxis recurre a C, Java y Perl, siendo así sencillo de aprender. El objetivo principal de este lenguaje es permitir a los desarrolladores web escribir dinámica y rápidamente páginas web generadas; aunque se puede hacer mucho más con PHP (php, 2011).

- JavaScript: JavaScript es un lenguaje de scripting multiplataforma y orientado a objetos. Es un lenguaje pequeño y liviano. Dentro de un ambiente de host, JavaScript puede conectarse a los objetos de su ambiente y proporcionar control programático sobre ellos.
- JavaScript contiene una librería estándar de objetos, tales como Array, Date, y Math, y un conjunto central de elementos del lenguaje, tales como operadores, estructuras de control, y sentencias. El núcleo de JavaScript puede extenderse para varios propósitos, complementándolo con objetos adicionales (MDN, 2009).

Marco de Trabajo

Un marco de trabajo o framework es un diseño abstracto orientado a objetos para un determinado tipo de aplicación, es un patrón arquitectónico que proporciona una plantilla extensible para un tipo específico de aplicaciones. Es un conjunto cohesivo de interfaces y clases que colaboran para proporcionar los servicios de la parte central e invariable de un subsistema lógico. El marco de trabajo diseñado posee una arquitectura orientada a componentes y está compuesto por cinco paquetes principales: Núcleo, Servicios, Persistencia, Componentes y Aplicación de Usuario. Cada paquete está compuesto por componentes y estos a su vez definen una interfaz de comportamiento con el objetivo de extender sus funcionalidades y adaptarse a posibles escenarios (Anglada, 2013).

Symfony 1.1.7 Symfony es un completo framework diseñado para optimizar, gracias a sus características, el desarrollo de las aplicaciones web, este separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación (Potencier, 2009).

- La capa de internacionalización que incluye Symfony permite la traducción de los datos y de la interfaz, así como la adaptación local de los contenidos.
- La capa de presentación utiliza plantillas y layouts que pueden ser creados por diseñadores HTML sin ningún tipo de conocimiento del framework. Los helpers incluidos permiten minimizar el código utilizado en la presentación, ya que encapsulan grandes bloques de código en llamadas simples a funciones.
- Los formularios incluyen validación automatizada y relleno automático de datos "repopulation", lo que asegura la obtención de datos correctos y mejora la experiencia de usuario.
- Los datos incluyen mecanismos de escape que permiten una mejor protección contra los ataques producidos por datos corruptos.
- La gestión de la caché reduce el ancho de banda utilizado y la carga del servidor.

- La autenticación y la gestión de credenciales simplifican la creación de secciones restringidas y la gestión de la seguridad de usuario.
- El sistema de enrutamiento y las URL limpias permiten considerar a las direcciones de las páginas como parte de la interfaz, además de estar optimizadas para los buscadores.
- El soporte de e-mail incluido y la gestión de APIs permiten a las aplicaciones web interactuar más allá de los navegadores.

ExtJS 3.4 De acuerdo a la definición, ExtJS es una librería JavaScript que permite construir aplicaciones complejas en Internet. Esta librería incluye:

- Componentes UI del alto performan y personalizables.
- Modelo de componentes extensibles.
- Un API fácil de usar.
- Licencias Open source y comerciales.

Una de las grandes ventajas de utilizar ExtJS es que permite crear aplicaciones complejas utilizando componentes predefinidos, así como un manejador de layouts similar al que provee Java Swing, gracias a esto provee una experiencia consistente sobre cualquier navegador, evitando el tedioso problema de validar que el código escrito funcione bien en cada uno (Firefox, IE, Safari, etc.) (antartec).

Como marcos de trabajo se utilizarán Symfony 1.1.7 y ExtJS 3.4 teniendo en cuenta que la programación del lado del servidor en SAGI se llevó a cabo con Symfony 1.1.7 y que ExtJS 3.4, en este caso utilizado para la programación de la interfaz, están diseñados basados en el patrón arquitectónico Modelo-Vista-Controlador, que facilita la separación de estos elementos posibilitando un balance entre Cliente – Servidor, de esta forma la carga de procesamiento se distribuye, permitiendo que el servidor, al tener menor carga, pueda manejar más clientes al mismo tiempo. También contienen una base de componentes amplia que permite la reutilización y extensión de los mismos, así como una extensa documentación.

Entorno de desarrollo integrado:

Un entorno de desarrollo integrado(IDE), es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI) (IBM, 2009).

NetBeans 8.0.2 Es un entorno de desarrollo gratuito y de código abierto que en el momento de escribir este artículo está en su versión 8.0.2. Permite el uso de un amplio rango de tecnologías de desarrollo tanto para escritorio, como aplicaciones Web, o para dispositivos móviles. Da soporte a las siguientes tecnologías, entre otras: Java, PHP, Groovy, C/C++, HTML5. Además, puede instalarse en varios sistemas operativos: Windows, Linux, Mac OS, etc. (genbetadev, 2014).

Este IDE de desarrollo se usa fundamentalmente por ser multilenguaje, pero además por ser de código abierto y gratuito. Por lo cual es la herramienta idónea para el desarrollo del módulo.

Sistema Gestor de Bases de Datos

PostgreSQL 9.3 PostgreSQL es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD y con su código fuente disponible libremente. Es el sistema de gestión de bases de datos de código (Martinez, 2013).

PostgreSQL utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando (Reingart, 2011).

PgAdmin III es una herramienta de código abierto para la administración de bases de datos PostgreSQL y derivados (EnterpriseDB, PostgresPlus, Advanced Server y GreenplumDatabase).

PgAdmin se diseña para responder a las necesidades de la mayoría de los usuarios, desde escribir simples consultas SQL hasta desarrollar bases de datos complejas. La interfaz gráfica soporta todas las características de PostgreSQL y hace simple la administración. Está disponible en más de una docena de lenguajes y para varios sistemas operativos, incluyendo Microsoft Windows, Linux, FreeBSD, Mac OSX y Solaris. Soporta versiones de servidores 7.3 y superiores. Versiones anteriores a 7.3 deben usar PgAdmin II (Reingart, 2011)

Servidor de aplicaciones:

Apache 2.2 Un servidor web es, como su nombre indica, un software instalado en el equipo con todas las condiciones necesarias para servir o entregar páginas web que le sean solicitadas por un navegador, asegurando que se muestren y representen todos los elementos necesarios para su correcto funcionamiento y visualización. Existen varios tipos de servidores web, Apache es un software de código abierto, libre de uso y totalmente configurable, es en este momento el más utilizado en la red, ya sea en plataformas Linux o Windows (apache, 2012).

El servidor está dirigido a servir a una gran cantidad de plataformas web que trabajan sobre sistemas operativos como Unix, Windows, Linux, Solaris, Novell NetWare, FreeBSD, Mac OS X, Microsoft Windows, OS / 2.

Patrón arquitectónico:

Editorial “Ediciones Futuro”
Universidad de las Ciencias Informáticas. La Habana, Cuba
seriecienf@uci.cu

Patrón arquitectónico: El módulo se desarrollará en Symfony lo que implica que se utilice el patrón arquitectónico Modelo-Vista-Controlador (MVC). Symfony está basado en un patrón clásico del diseño web conocido como arquitectura MVC, que está formado por tres niveles:

El Modelo representa la información con la que trabaja la aplicación, es decir, su lógica de negocio.

La Vista transforma el modelo en una página web que permite al usuario interactuar con ella.

El Controlador se encarga de procesar las interacciones del usuario y realiza los cambios apropiados en el modelo o en la vista.

La arquitectura MVC separa la lógica de negocio (el modelo) y la presentación (la vista) por lo que se consigue un mantenimiento más sencillo de las aplicaciones. El controlador se encarga de aislar al modelo y a la vista de los detalles del protocolo utilizado para las peticiones (HTTP, consola de comandos, email). El modelo se encarga de la abstracción de la lógica relacionada con los datos, haciendo que la vista y las acciones sean independientes de, por ejemplo, el tipo de gestor de bases de datos utilizado por la aplicación (Potencier, 2015).

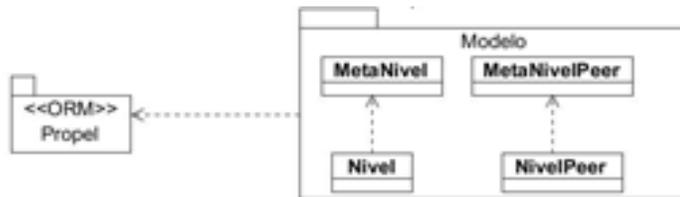
Patrones GRASP

Los patrones GRASP describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones.

GRASP es un acrónimo que significa General Responsibility Assignment Software Patterns (patrones generales de software para asignar responsabilidades) El nombre se eligió para indicar la importancia de captar (grasping) estos principios, si se quiere diseñar eficazmente el software orientado a objetos (GAMMA, y otros, 2009).

Experto:

Asignar una responsabilidad al experto en información, la clase que tiene la información necesaria para realizar la responsabilidad. Este es uno de los más utilizados, puesto que Propel es la librería externa que utiliza Symfony para realizar su capa de abstracción en el modelo, encapsula toda la lógica de los datos y son generadas las clases con todas las funcionalidades comunes de las entidades. Propel como ORM crea cuatro clases: dos clases de acceso a datos que trabajan directamente con la base de datos y dos de abstracción de datos que son las que tienen los atributos necesarios para realizar dicha función.



Patrón Experto.

Bajo Acoplamiento: Es la medida de cuánto una clase está conectada (tiene conocimiento) de otras clases. Es un patrón evaluativo: un bajo acoplamiento permite que el diseño de clases sea más independiente. La clase Actions hereda únicamente de sfActions para alcanzar un bajo acoplamiento de clases. Las clases que implementan la lógica del negocio y de acceso a datos se encuentran en el modelo, las cuales no tienen asociaciones con las de la vista o el controlador, lo que proporciona que la dependencia en este caso sea baja.

```

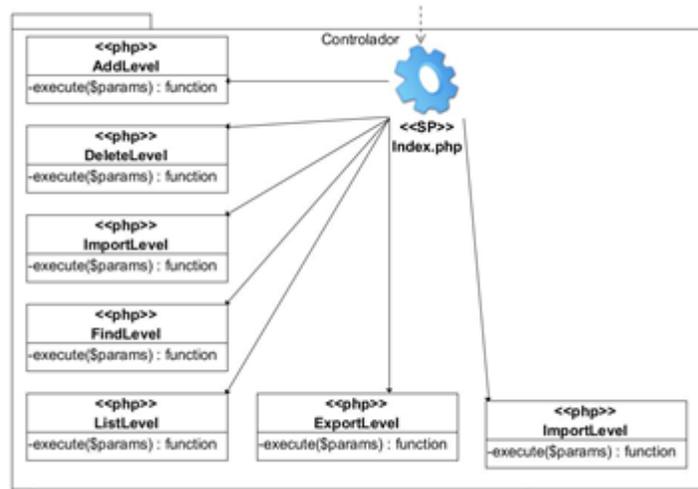
class addLevelsAction extends MySIGAction {
    public function execute($request) {
        try {
            $level = new metaNivel();
            $level->setNombre($request->getParameter('nombre'));
            $level->setOrden($request->getParameter('orden'));
            $level->setSubstr($request->getParameter('substr'));
            $level->setLongstr($request->getParameter('longstr'));
            $level->setDescripcion($request->getParameter('descripcion'));
            $level->setAcumuladoNombre($request->getParameter('acumulado'));
            $level->setEnMaestro($request->getParameter('en_maestro'));
            $level->setAceptaColumna($request->getParameter('acepta_columna'));
            $level->setTorigen($request->getParameter('torigen'));
            $level->setClasificadorId($request->getParameter('codclasificador'));
            $level->setTdato($request->getParameter('tdato'));

            $level->save();

            return $this->renderText("{\"success:true\"}");
        } catch (Exception $e) {
            if ($e->getCode() == 1)
                return $this->renderText("{\"success:false, errors:{message:'\" . $e->getMessage() .
            else {
    
```

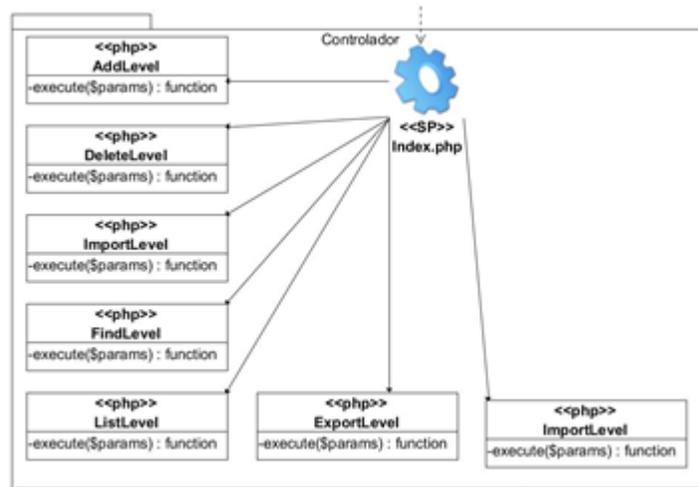
Patrón Bajo Acoplamiento.

Alta Cohesión: Define lo relacionadas que están las responsabilidades de una clase, o una clase con responsabilidades altamente relacionadas y que no lleva a cabo gran cantidad de trabajo. La clase Index.php interactúa con las 7 clases Actions para recuperar, crear y almacenar objetos.



Patrón Alta Cohesión.

Controlador: Es un objeto que no pertenece a la interfaz de usuario, responsable de recibir o manejar un evento del sistema. Un Controlador define el método para la operación del sistema. Todas las peticiones Web son manipuladas por un solo controlador frontal (sfActions), que es el punto de entrada único de toda la aplicación en un entorno determinado. En este caso el “index.php” escucha las peticiones que vienen desde una URL, luego maneja la acción requerida para satisfacer la petición realizada

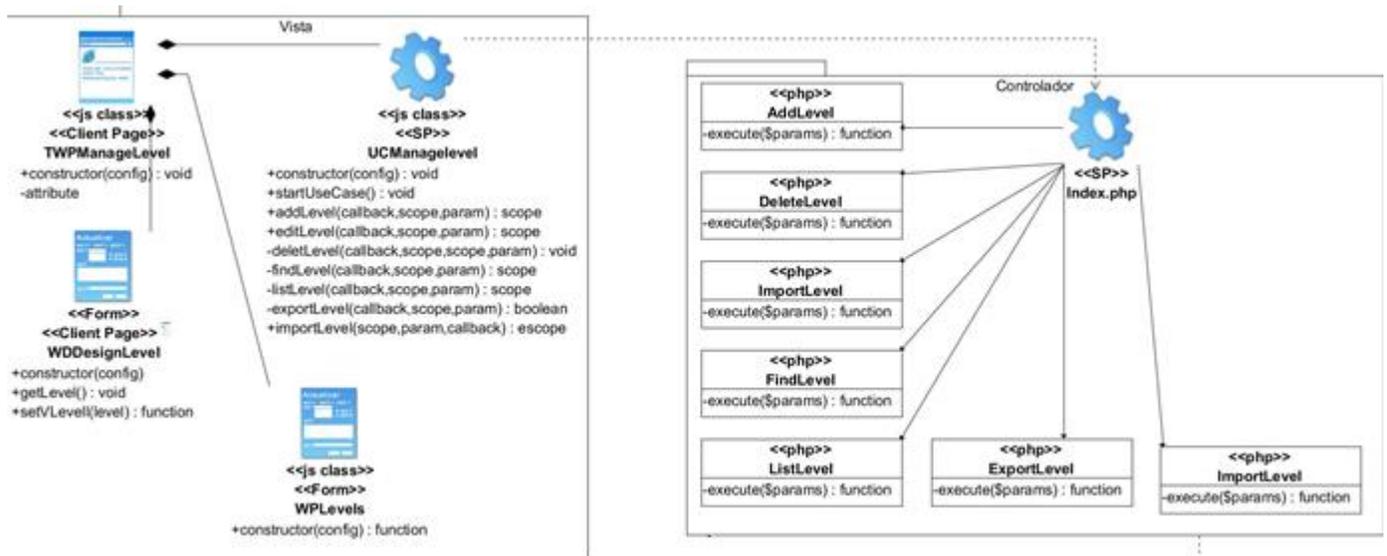


Patrón Controlador

Patrones GoF

Los patrones GoF (Gang of Four) se utilizan en situaciones frecuentes. Debido a que se basan en la experiencia acumulada al resolver problemas reiterativos. Ayudan a construir software basado en la reutilización, a construir clases reutilizables (JACOBSON, y otros, 2004).

Fachada (estructural): Se requiere una interfaz común para un conjunto de implementaciones o interfaces dispares. Se define la clase UCMangeLevel como el único punto de conexión de la clase TWPMangeLevel, este objeto fachada presenta una única interfaz y es responsable de colaborar con los clientes (Controlador de fachada).



Patrón Fachada.

Conclusiones

- La implementación de las funcionalidades del módulo permitió obtener una aplicación informática que responde a las necesidades del cliente definidas durante la etapa de análisis.
- Las pruebas diseñadas y aplicadas al módulo garantizaron el correcto funcionamiento del mismo, detectando no conformidades que fueron solucionadas posteriormente.
- El desarrollo del Módulo para la Gestión de Rutinas para el SAGI permitió la creación de las mismas en el SAGI posibilitando así un mayor control del cumplimiento de los reportes y que los mismos puedan ser entregados y analizados en el tiempo estimado.

Referencias

- ACOBSON, Ivar y BOOCH, Grady, RUMBAUGH, James. 2000. El Proceso Unificado. 2000.
- Agualló, P. 2010. Desarrollo Cliente / servidor: ubicación de las reglas de negocio. Desarrollo Cliente / servidor: ubicación de las reglas de negocio. [En línea] 2010. [Citado el: 15 de febrero de 2015.] <http://www.ctv.es/USERS/pagullo/arti/esbr/esbr.htm>.
- antartec. Desarrollo en Web. Desarrollo en Web. [En línea] <http://blogs.antartec.com/desarrolloweb/tag/extjs/>.
- apache. apache.org 2.2 con PHP. apache.org 2.2 con PHP. [En línea] [Citado el: 26 de octubre de 2015.] <https://csi2altair.wikispaces.com/Apache.org+2.2+con+PHP>.
- Bruegge, B. Y Dutoit, A. 2002. Ingeniería de Software Orientado a Objetos. 2002.
- Cornejo, José Enrique González. 2008. DocIRS. DocIRS. [En línea] enero de 2008. [Citado el: 24 de octubre de 2015.] <http://www.docirs.com/uml.htm>.
- Fabien Potencier, François Zaninotto. Symfony, la guía definitiva. Symfony, la guía definitiva. [En línea] [Citado el: 20 de noviembre de 2015.] http://librosweb.es/libro/symfony_1_4/.
- fergarcia. 2013. fergarcia. fergarcia. [En línea] 25 de enero de 2013. [Citado el: 14 de diciembre de 2015.] <https://fergarcia.wordpress.com/2013/01/25/entorno-de-desarrollo-integrado-ide/>.
- GAMMA, E., HELM, R. y JOHNSON, R. y VLISSIDES, J. Patrones de diseño. Patrones de diseño. Patrones de diseño. [En línea] vico, Patrones de diseño. <http://www.vico.org/pages/PatronsDisseny.html>.
- genbetadev. 2014. genbetadev: Desarrollo de Software. genbetadev: Desarrollo de Software. [En línea] 9 de enero

de 2014. [Citado el: 15 de diciembre de 2015.] <http://www.genbetadev.com/herramientas/netbeans-1>.

Hernandez Barrios, Carmelo Jose. 2014. slideshare. [En línea] 2014. [Citado el: 4 de Febrero de 2015.] <http://es.slideshare.net/carmeloh2/metodologa-open-up-39321348>.

IBM. 2009. IBM Knowledge Center. IBM Knowledge Center. [En línea] IBM, diciembre de 2009. [Citado el: 24 de octubre de 2015.] [/www-01.ibm.com/support/knowledgecenter/SSEPGG_8.2.0/com.ibm.db2.udb.doc/ad/c0009415.htm?lang=es](http://www-01.ibm.com/support/knowledgecenter/SSEPGG_8.2.0/com.ibm.db2.udb.doc/ad/c0009415.htm?lang=es).

Marin, Marvin David Arias. 2008. Catedra de Programación. Catedra de Programación. [En línea] 16 de octubre de 2008. [Citado el: 26 de octubre de 2015.] <http://catedraprogramacion.forosactivos.net/t83-definicion-de-lenguaje-de-programacion-tipos-ejemplos>.

Martinez, Rafael. 2009-2013. PostgreSQL-es. PostgreSQL-es. [En línea] 2009-2013. [Citado el: 15 de diciembre de 2015.] http://www.postgresql.org/es/sobre_postgresql.

MDN. MDN. MDN. [En línea] [Citado el: 26 de octubre de 2015.] <https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Introducci%C3%B3n>.

Paradigm. Paradigm, 2010. Paradigm, 2010. [En línea] [Citado el: 15 de noviembre de 2015.] <http://www.visual-paradigm.com/aboutus/newsreleases/vpuml80.jsp>.

php. php. php. [En línea] [Citado el: 26 de octubre de 2015.] <http://php.net/manual/es/preface.php>.

Potencier, Fabien. 2015. librosweb. [En línea] 2015. [Citado el: 6 de Febrero de 2016.] http://librosweb.es/libro/symfony_1_1/capitulo_2/el_patron_mvc.html.

PRESSMAN, Roger. 2009. Ingeniería del Software. Un enfoque práctico. s.l. : McGraw-Hill/Interamericana, 2009.

Ramón Alexander Anglada Martínez, Alain Abel Garófalo Hernández. 2013. Revista Cubana de Ciencias Informáticas. Revista Cubana de Ciencias Informáticas. [En línea] abr-jun. de 2013. [Citado el: 26 de octubre de 2015.] http://scielo.sld.cu/scielo.php?pid=S2227-18992013000200006&script=sci_arttext. ISSN 2227-1899.

Reingart, Mariano. postgresql. postgresql. [En línea] [Citado el: 26 de octubre de 2015.] <http://www.postgresql.org.ar/trac/wiki/PgAdmin?format=pdfarticle>.

Sparx. 2000-2007. Sparx Systems. Sparx Systems. [En línea] Sparx Systems, 2000-2007. [Citado el: 21 de enero

de 2016.] http://www.sparxsystems.com.ar/resources/tutorial/use_case_model.html. ISBN.

Tecnología. SlideShare. SlideShare. [En línea] [Citado el: 26 de octubre de 2015.]
<http://es.slideshare.net/guestf131a9/herramientas-case>.

UML. UML. [En línea] [Citado el: 14 de octubre de 2015.] <http://www.uml-diagrams.org/>.

wordpress. 2010. wordpress.com. wordpress.com. [En línea] agosto de 2010. [Citado el: 24 de octubre de 2015.]
<https://tombasededatos.wordpress.com/2010/08/28/2-1-definicion-de-un-modelo-de-datos/>.