

Tipo de artículo: Artículo original
Temática: Informática Jurídica
Recibido: 30/10/15 | Aceptado: 13/01/2016

Diseño e implementación de los Flujos alternos del proceso Ejecutivo del SITPC

Design and implementation of the Alternative flows of the Executive process of SITPC

Juan Alberto Pereira Delgado¹

¹ Dirección de Redes y Servicios. Universidad de las Ciencias Informáticas, Carretera a San Antonio de los Baños, km 2 ½, La Lisa, Torrens, La Habana. Cuba. japereira@uci.cu

* Autor para correspondencia: japereira@uci.cu

Resumen

La información es uno de los elementos fundamentales para cualquier proceso empresarial, pero cada vez se hace más difícil mantener las herramientas de gestión de la información al margen del desarrollo tecnológico. Sin embargo, la brecha tecnológica no ha sido un impedimento para que el país pueda avanzar en el mundo de la informática; tomando como alternativa la creación de la Universidad de las Ciencias Informáticas, la cual entre otras actividades desarrolló un sistema de gestión orientado a satisfacer las necesidades informáticas de los Tribunales Populares Cubanos. Una de las materias que se desarrolla en los tribunales es la Económica, en la cual se efectúa, entre otros, el proceso Ejecutivo. De dicho proceso se realizó una primera versión, que no resultó ser eficiente, debido a que solo se implementaba el Flujo básico que debía seguir el mismo. Por tal motivo, se pretende documentar cómo se implementaron los Flujos alternos para dar completamiento al módulo Ejecutivo, lográndose así una aplicación de calidad. Como resultado de la utilización de herramientas, lenguajes de programación, patrones de diseño y una fuerte arquitectura, elementos imprescindibles para mantener un sistema estructurado y funcional, se obtuvo un módulo íntegro del proceso Ejecutivo, el cual fue integrado de manera satisfactoria. Las pruebas aplicadas para comprobar la excelencia y el buen funcionamiento del sistema resultante, demostraron que el mismo cumplía con las necesidades del cliente, además fue capaz de resolver todas las deficiencias que inicialmente afectaban el correcto desarrollo del módulo.

Palabras clave: alternos; tribunales; Ejecutivo; Económico; subsistema.

Abstract

Information is one of the key elements for any business process, but it is becoming more difficult to maintain the information management tools outside the technology development. However, the technology breach has not been an impediment to the country to advance in the world of computing; taking alternatively the creation of the University of Informatics Sciences, which among other activities developed a management system oriented to satisfy the informatics needs of the Popular Tribunals of Cuba. One of the subjects is developed in the Economic court, in which the Executive process is carried out, among others. In this process, a first version proved not to be efficient, because only the Basic flow that should follow the same was implemented. Therefore, it is intended to document how the Alternative flows were implemented to give completion to the Executive module, thus achieving a quality application. As a result of the use of tools, programming languages, design patterns and a strong architecture, essential to maintain a structured and functional system, an integral module of the Executive process was obtained and successfully integrated. The tests used to check excellence and good performance of the resulting system, demonstrated that it met the needs of the customer, also was able to resolve all deficiencies that initially affected the smooth operation of the module.

Keywords: *alternative; tribunals; Executive; Economic; module.*

Introducción

El actual siglo está marcado por un rápido desarrollo de la tecnología en diversas ramas de la actividad humana como: la ciencia, la educación, las comunicaciones, entre otras. El desarrollo tecnológico ha sido un factor determinante de cambio para procesos como la gestión de la información, considerado piedra angular para el éxito de las empresas. Una correcta gestión de la información asegura su integridad, disponibilidad y confidencialidad, contribuyendo a la eficacia, eficiencia y productividad de las organizaciones.

En Cuba, el proceso de gestión de la información ha evolucionado muy lentamente, sin embargo, han sido muchas las alternativas puestas en práctica para mejorar las actividades que tienen una marcada dependencia de la información. Con el objetivo de “*producir aplicaciones y servicios informáticos a través del uso de las nuevas tecnologías, además de servir de soporte a la industria cubana de la informática*” (UCI, 2012), surge la Universidad de las Ciencias Informáticas. En dicha institución radican diversos centros de desarrollo de software, encargados de informatizar servicios que abarcan líneas vitales para el desarrollo del país. Uno de ellos es el Centro de Gobierno Electrónico (CEGEL), el cual tiene la misión de satisfacer necesidades de clientes gubernamentales mediante el desarrollo de productos, servicios y soluciones informáticas de alta confiabilidad (Laboratorio de Gestión de Proyectos, 2010).

Actualmente, una de las entidades a la cual CEGEL le está brindando servicios es al Tribunal Supremo Popular. En las diferentes instancias de esta organización (tribunales municipales y provinciales) se han estado presentando un conjunto de problemas que han estado afectando la eficiencia en la realización de sus procesos, algunos de ellos son: el trabajo manual en los procedimientos provoca la existencia de duplicaciones en la radicación¹ de los expedientes, además los documentos oficiales contienen borrones y tachaduras, efecto que incide negativamente sobre la estética de los mismos; sumando también la desactualización de los registros debido a la demora en la introducción de datos en los libros.

Con la intención de resolver estas deficiencias el Tribunal Supremo decide, en coordinación con el centro CEGEL, desarrollar un sistema de gestión con el objetivo de informatizar los procesos que se realizan en cada una de las materias que componen los tribunales cubanos, surgiendo de esta manera el Sistema de Informatización para la Gestión de los Tribunales Populares Cubanos (SITPC en lo adelante). Dicho sistema desarrolla cinco subsistemas: Administrativo, Económico, Laboral, Penal y Civil; donde cada uno de ellos define internamente un conjunto de módulos. El subsistema Económico, al cual se hará referencia en lo adelante, implementa cuatro módulos: Ordinario, Casación, Revisión y Ejecutivo; siendo este último el centro de la presente investigación.

El módulo Ejecutivo tiene como principal tarea informatizar los trámites concernientes a este proceso dentro de la materia Económica, dicho módulo se compone de la realización del Flujo básico en conjunto con los Flujos alternos; si algunos de los procedimientos que integran ambos flujos no son desarrollados, no se puede obtener un sistema funcionalmente íntegro para el proceso Ejecutivo. Sin embargo, como resultado de la primera iteración realizada, según el plan de iteraciones inicialmente elaborado para el módulo, solo fueron implementados los procedimientos correspondientes al Flujo básico, no siendo así para los Flujos alternos. Al no estar implementadas dichas funcionalidades, el sistema desarrollado hasta el momento no es capaz de realizar las disposiciones² sobre los escritos que se insertaban durante el proceso, provocando en adición la interrupción o retraso en la continuación de los trámites posteriores a estos procedimientos; además limitaba a seguir un único flujo la realización de los procedimientos. Por otro lado, no son gestionados en el sistema algunos de los procedimientos con los que finaliza al proceso Ejecutivo, los cuales son determinantes para que el mismo se efectúe correctamente hasta su terminación.

Debido a que la primera iteración implementada del módulo Ejecutivo solo satisface una parte de las necesidades del cliente, puesto que no cuenta con la implementación de las funcionalidades correspondientes a los Flujos alternos,

1 Proceso en el cual se le asigna un número al expediente y se asienta en el libro.

2 Proceso de admitir, subsanar o rechazar un escrito presentado, por una parte.

necesarios para la completitud del proceso, se propone desarrollar una segunda iteración del módulo Ejecutivo según el plan de iteraciones inicial, para implementar las funcionalidades relativas a los Flujos alternos que no están presentes en la primera iteración, permitiendo completar el módulo.

Partiendo de lo anterior, se definen los siguientes objetivos:

- Transformar los requisitos funcionales a un lenguaje artefactual para identificar las relaciones entre las partes del sistema, mediante el modelo de diseño.
- Obtener un sistema funcional para dar solución a los requisitos definidos previamente para los Flujos alternos, mediante la implementación de casos de uso del sistema.
- Realizar las pruebas de software necesarias para verificar y validar los resultados obtenidos.

Materiales y métodos

El SITPC se desarrolló utilizando la metodología RUP (*Rational Unified Process*). Cuenta con una arquitectura Modelo-Vista-Controlador (MVC) basada en el estilo arquitectónico de llamada y retorno; y a su vez se utilizan los patrones de diseño orientados a objetos GoF (Decorador y Método Fábrica), Grasp (Experto, Creador, Controlador, Bajo acoplamiento y Alta Cohesión) (*Larman, 1999*) e inyección de dependencias (*symfony.es. 2009*).

Para el modelado de los diagramas de clases, de comunicación y de despliegue y del modelo de datos, del subsistema en cuestión, se utilizó la herramienta CASE Visual Paradigm v8.0 haciendo uso del lenguaje UML v2.0 (*Larman, 1999*). Durante el desarrollo del subsistema se utilizó como entorno de desarrollo NetBeans IDE v7.4 (netbeans.org, 2000) y como sistema gestor de base de datos PostgreSQL v9.2 (*PostgreSQL, 2010*). Para el control de las versiones fue utilizado SubVersion v1.5 (*Collins-Sussman, et al., 2008*) y como servidor web Apache v2.2 (*Apache, 2012*). Symfony v2.1 (*symfony.es. 2009*) fue el marco de trabajo utilizado, el cual utiliza a su vez el motor de plantillas Twig y los marcos de trabajo Bootstrap v2.1 y jQuery v1.8. Se integró al sistema el componente de seguridad AcaXia (*Ing. Roque Hernández, 2013*) y fue utilizado el mapeador objeto-relacional (ORM) Doctrine v2.0 (*Eguiluz, 2011*). Fueron empleados los lenguajes PHP v5.3 (*The PHP Group, 2001-2014*), HTML 5 (*W3C Nightly, 2013*), CSS 3 (*CSS, 2012*) y JavaScript 3 (*Eguiluz, 2007*).

Fueron empleadas la prueba de caja blanca haciendo uso de la técnica de camino básico y la prueba de caja negra, mediante la técnica de partición equivalente, respectivamente. Así como la aplicación de determinadas métricas orientadas al diseño y a las clases con el objetivo de demostrar cualitativa y cuantitativamente cuán eficiente es la aplicación. La aplicación de estas pruebas garantizó que el sistema quedara con la calidad requerida.

Resultados y discusión

En el subsistema, siguiendo la arquitectura Modelo-Vista-Controlador empleada por Symfony v2.1, utiliza para el diseño de la aplicación 4 capas: de presentación, de negocio, de abstracción de datos y de datos sin que se viole la seguridad y la lógica del sistema. De manera transversal a las capas se encuentran las entidades del dominio y los componentes útiles para el tratamiento de la seguridad (AcaXia), así como las excepciones, inyección de dependencias, entre otros.

En la capa de presentación se encuentran las interfaces de los usuarios y la lógica de control. Mediante las interfaces se muestra o capta la información utilizando componentes propios de las vistas (formularios, grids, mensajes de estado o confirmación, entre otros); a la vez que recibe las funcionalidades que le son enviadas desde el controlador y funciona mediante el principio de la arquitectura cliente-servidor. La capa de negocio gestiona y procesa las peticiones recibidas desde el nivel superior, para su posterior envío de la respuesta. Esta capa es la encargada de proveer a las interfaces la información que soliciten a través de las clases gestoras. La capa de abstracción de datos tiene la función de gestionar las peticiones de consultas a la base de datos, recibidas desde la capa de negocio. En esta capa se encuentran las clases repositorios y entidades. Y la capa de datos ofrece las herramientas necesarias para manipular y almacenar la información con la que trabajará la aplicación (Sistema Gestor de Base de Datos PostgreSQL v9.2); almacenando los esquemas, tablas, funciones y procedimientos necesarios para su posterior utilización.

El modelo de datos elaborado para el modulo Ejecutivo, dentro del subsistema Económico consta de un total de 17 tablas, de ellas 5 son nomencladores y los 12 restantes son de información referente a los objetos. Las entidades del modelo representan, en su mayoría, una tabla en la base de datos que permiten manejar la información de los objetos.

Los diagramas de comunicación elaborados se diseñaron siguiendo una notación básica, identificándose los mismos por la expresión: *numeroMensaje: mensaje (parámetro: tipoparametro): tipoRetorno* (Hernández Orallo, 2013). La siguiente figura es un fragmento de los casos de uso (CU) correspondientes a los Flujos alternos en la que muestra la manera en que se realizaron todos los diagramas de comunicación:

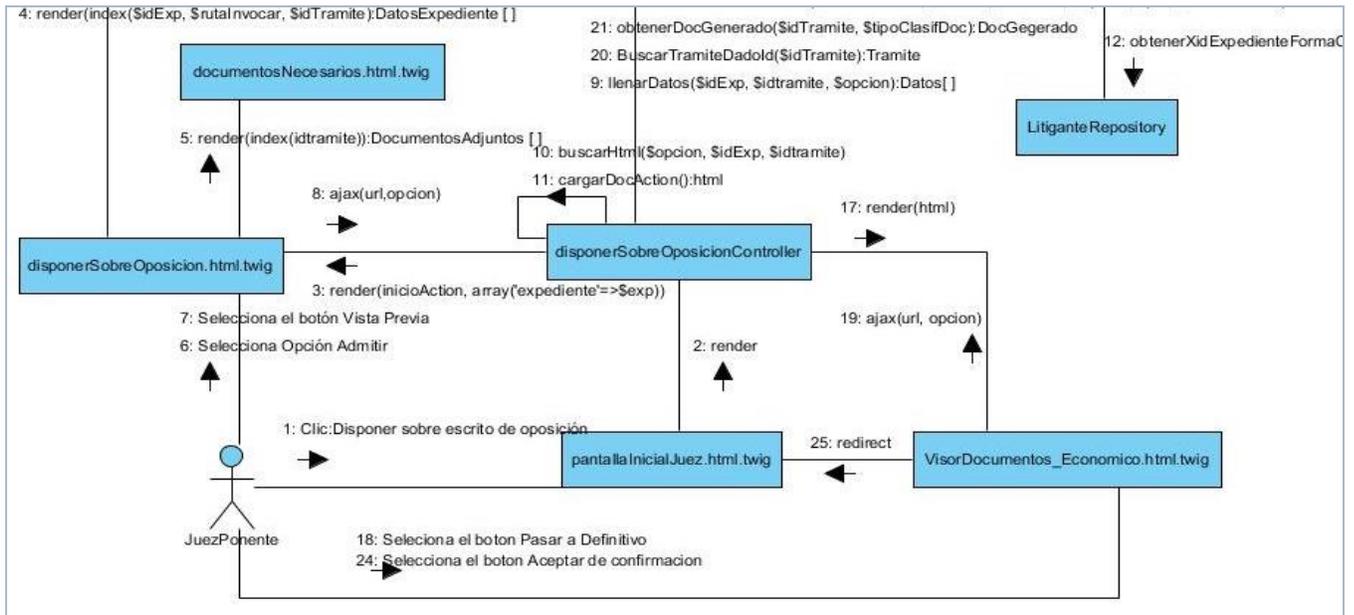


Figura 1. Fragmento del diagrama de comunicación para el CU Disponer sobre escrito de oposición.

Luego de finalizar el diseño de los diagramas de comunicación, se realizó como paso siguiente la modelación de los diagramas de clases. Este tipo de diagramas describe gráficamente las entidades que intervienen en el desarrollo de la aplicación, así como la relación que existe entre ellas. Cada entidad o clase refleja las características de un objeto, así como su comportamiento a través de operaciones o métodos.

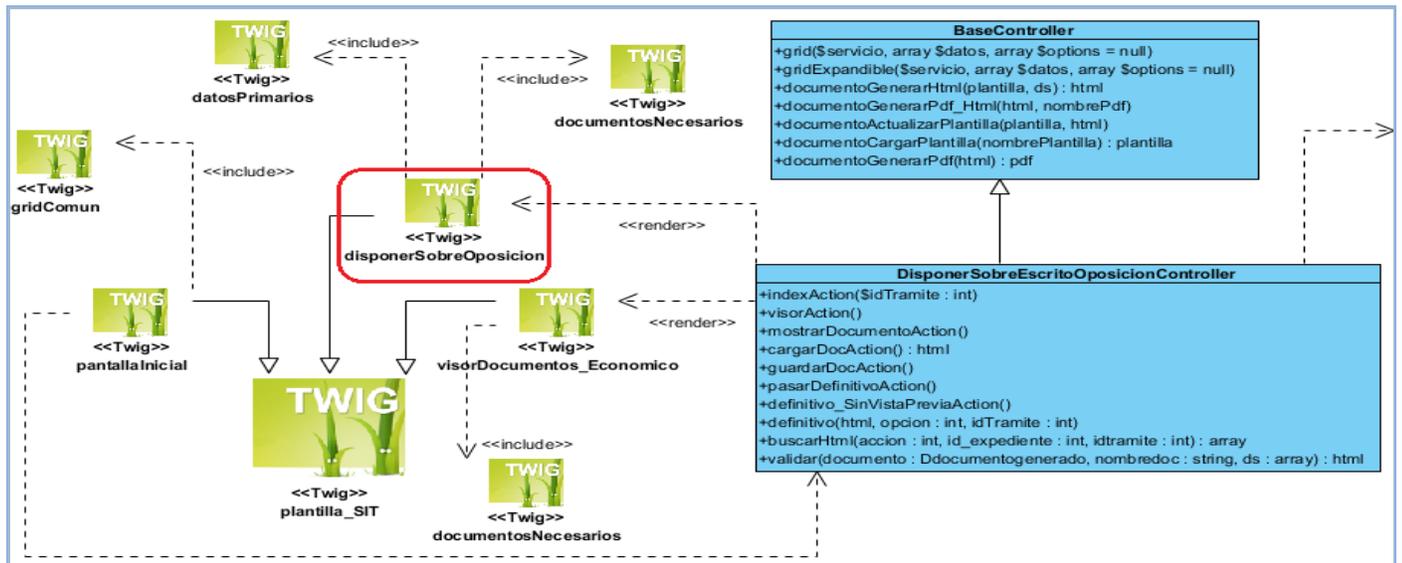


Figura 2. Fragmento del diagrama de clases del CU Disponer sobre escrito de oposición.

En el diagrama que se presenta en la figura 2, se puede observar cómo interactúan las interfaces del sistema, la interfaz principal del caso de uso (enmarcada en rojo) y la clase controladora, así como la relación que existe entre esta y la clase de la que hereda.

Para el SITPC se definió, específicamente orientados al modelo de despliegue, una serie de requisitos no funcionales (Ing. Roque Hernández, 2013 pág. 12); a lo cual se suma el uso de impresoras conectadas a las estaciones de trabajo, con el propósito de brindar servicios básicos internos y externos a la entidad. El servidor de base de datos del Tribunal Supremo Popular almacenará la información de todas las instancias, preferiblemente en horario no laboral; también existirá otro servidor que mantiene una réplica de la misma. Los servidores web de las instancias municipales y provinciales contendrán todo lo relacionado con la aplicación. Siguiendo lo expresado anteriormente, la interfaz de usuario ofrece una vista agradable al mismo, y permite una manipulación sencilla y transparente de los datos que en ella se muestran, de manera que se eviten la introducción de errores en el sistema y el tecleo sea mínimo.

Figura 3. Vista principal del CU Disponer sobre escrito de oposición.

Durante la validación del módulo fue utilizada la métrica Tamaño de clase (TC) que evalúa los atributos de Responsabilidad, Complejidad de implementación y Reutilización; arrojando los siguientes resultados:

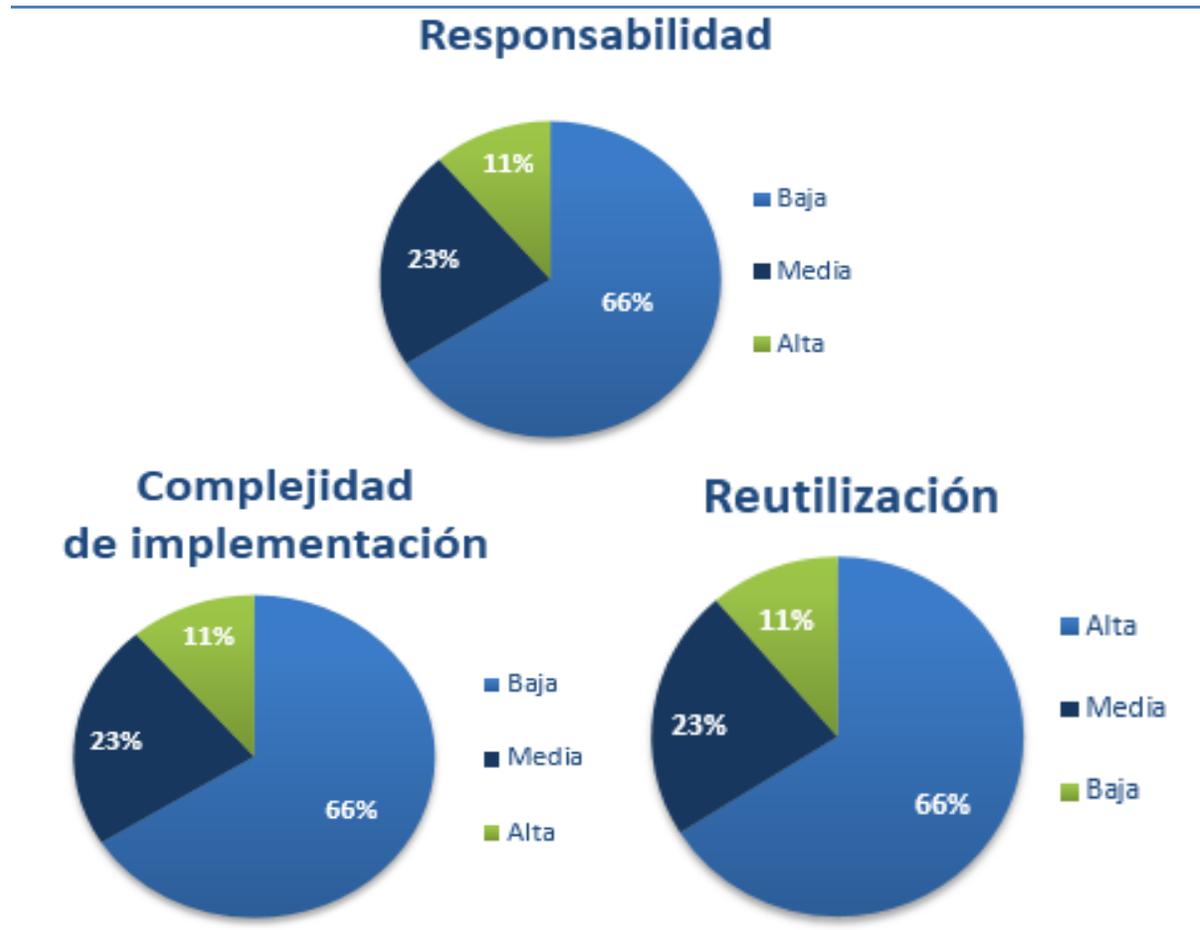


Figura 4. Resultado obtenido de aplicar la métrica TC.

La métrica Relaciones entre clases (RC) evalúa los atributos de calidad Acoplamiento, Complejidad de mantenimiento y Reutilización, sus resultados fueron los siguientes:

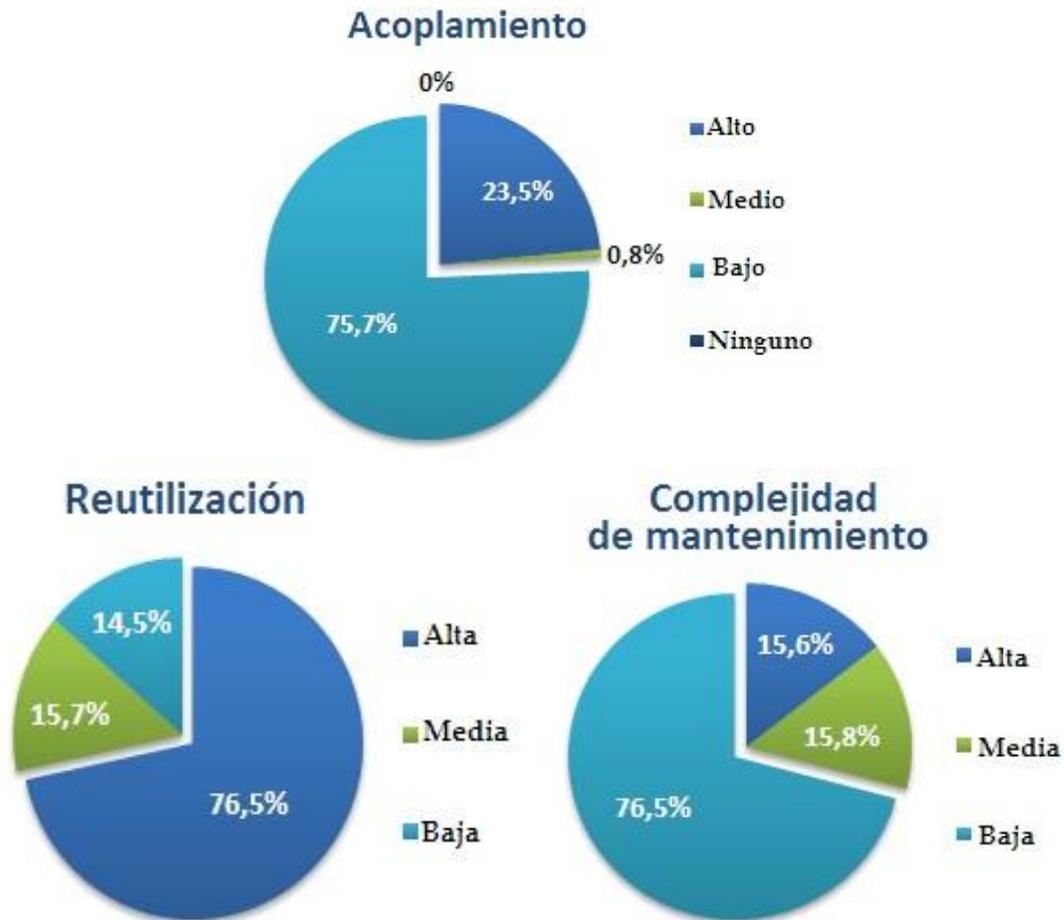


Figura 4. Resultado obtenido de aplicar la métrica RC.

Las métricas mencionadas anteriormente fueron aplicadas para corroborar el diseño orientado a objetos. Para evaluar el diseño orientado a clases fueron utilizadas las métricas Falta de cohesión en métodos (FCM) y Árbol de profundidad de herencia (APH), respectivamente. Los resultados de la primera (FCM) arrojaron que el valor promedio es 2, sin embargo, puede alcanzar un valor máximo de 4, lo que indica que: (1) existe una buena subdivisión de clases, manteniéndose lo menos acopladas posibles por atributos; por consiguiente (2) facilita el mantenimiento de las mismas, en tanto que (3) disminuye su complejidad y las oportunidades de que ocurran errores durante el proceso de desarrollo.

Los resultados de APH demuestran que: (1) el valor es aceptable, pues la longitud de las clases está muy por debajo del valor óptimo que proponen Chidamber y Kemerer³ para esta métrica (Stephen, 2002). Por otra parte, se puede afirmar que (2) la probabilidad de que las clases descendientes hereden la mayor cantidad de métodos posibles es baja; (3) provocando que el diseño tenga un alto nivel de facilidad de mantenimiento.

Luego de aplicada la técnica del camino básico (correspondiente a las pruebas de caja blanca) sobre los métodos de las clases controladoras, se detectaron las siguientes no conformidades (NC):

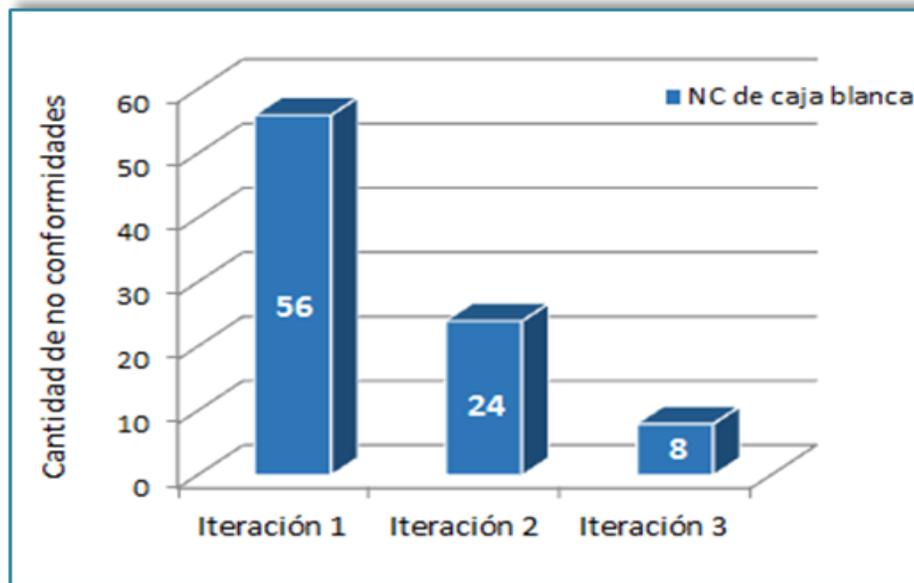


Figura 5. Cantidad de no conformidades encontradas en la prueba de caja blanca.

La cantidad encontrada en la última iteración representa menos del 30% de la sumatoria de las no conformidades encontradas en las iteraciones anteriores, cumpliéndose de esta forma el criterio de parada propuesto; las cuales fueron corregidas y se logró asegurar el cumplimiento del proceso de mejora del código.

Las pruebas de integración fueron realizadas para verificar la correcta coherencia entre los diferentes módulos o casos de uso y resultaron satisfactorias. Posteriormente como apoyo al proceso de verificación del sistema, el Departamento de Calidad del centro CEGEL realizó las pruebas necesarias para comprobar el correcto funcionamiento de las operaciones implementadas (prueba de comportamiento o caja negra), obteniéndose los siguientes resultados:

3 Dr. Shyam R. Chidamber y Dr. Chris F. Kemerer son profesores adjuntos al Dpto. de Negocio Internacional de la Universidad de Pittsburgh.

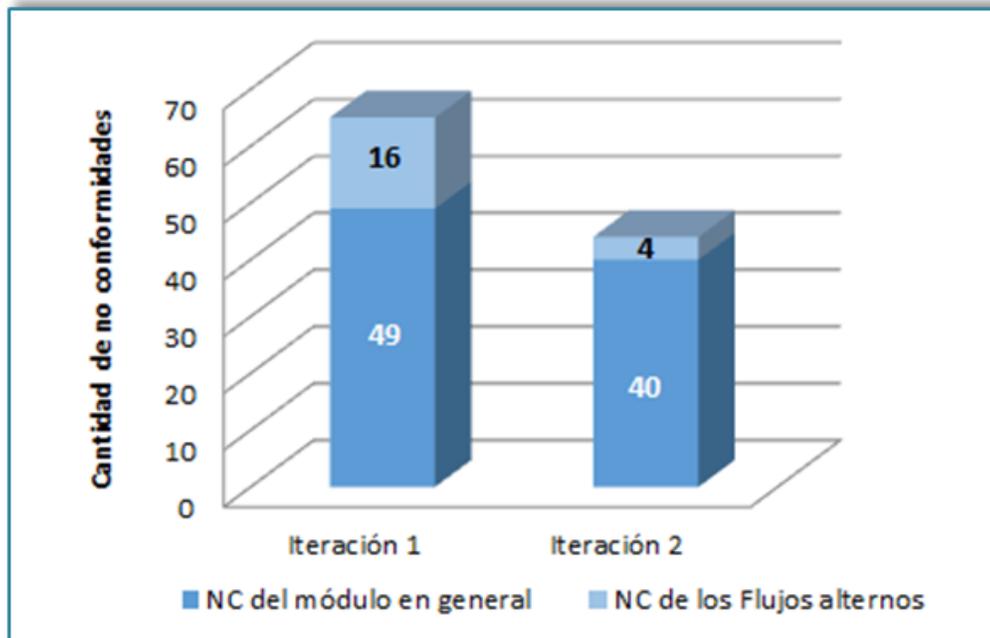


Figura 61. Cantidad de no conformidades encontradas por iteración, en la fase de prueba externa de los Flujos alternos.

Luego de realizadas dos iteraciones, se encontraron en cada una de ellas un determinado número de NC, las mismas representan los Flujos alternos y su integración con el Flujo básico (Ing. Maza Oval, et al., 2014), se puede observar que la cantidad de errores encontrados en el sistema, correspondientes a las funcionalidades de los Flujos alternos, no ofrecen valores relevantes con respecto a los resultados obtenidos de todo el módulo.

Después de haber sido resueltas todas las NC encontradas en la última revisión, se puede concluir que las funcionalidades implementadas para los procedimientos alternos del proceso Ejecutivo cumplen con los requisitos especificados en la fase de análisis. Además, la cantidad de NC encontradas mediante el método de caja negra, tanto a nivel interno como externo, demuestra un elevado nivel de efectividad en cuanto al diseño y ejecución de los casos de prueba, y la implementación de las funcionalidades a las que estos se asocian.

Conclusiones

Mediante el presente trabajo se ha documentado cómo se realizó la implementación de los Flujos alternos del módulo Ejecutivo del subsistema Económico del SITPC. Luego de finalizar la investigación de manera exitosa, se arribaron a las siguientes conclusiones:

- ❖ La metodología utilizada como herramienta de planificación y control del proyecto condujo durante todo el proceso las tareas y actividades necesarias para dar cumplimiento al desarrollo de los Flujos alternos.
- ❖ El uso de patrones de diseño permitió transformar los requisitos funcionales a un lenguaje artefactual obteniéndose un modelo de diseño con calidad.
- ❖ En la etapa de implementación se codificaron los requisitos establecidos para los Flujos alternos, obteniéndose un sistema funcional e íntegro del módulo Ejecutivo.
- ❖ Mediante el uso de las métricas de diseño seleccionadas se pudo validar que la aplicación posee un elevado nivel de reutilización, propicio para la realización sencilla del mantenimiento.
- ❖ Las pruebas realizadas al sistema fueron satisfactorias, de manera que se comprobó que las funcionalidades implementadas cumplen con los requisitos definidos.

De esta manera, quedan satisfechas las necesidades del cliente. El sistema se encuentra desplegado, lográndose así un hito en la informatización de entidades públicas, en este caso los Tribunales Populares Cubanos; y puesto que sistemas anteriores no satisfacían los requerimientos planteados se puede argumentar que las salas de lo Económico de dichos tribunales cuentan con un recurso valioso para poder resolver las necesidades del pueblo, así como las suyas propias.

Referencias

ALEXANDER, CHRISTOPHER, ISHIKAWA, SARA Y SILVERSTEIN, MURRAY. 1977. A Pattern Language. s.l.: Oxford University Press, 1977. ISBN: 0-19-501919-9

APACHE, SOFTWARE FOUNDATION. 2012. What is Apache. [En línea] 17 de Enero de 2012. [Citado el: 2 de Diciembre de 2013.] http://wiki.apache.org/httpd/FAQ#What_is_Apache.3F.

COLLINS-SUSSMAN, BEN, W. FITZPATRICK, BRIAN Y PILATO, C. MICHAEL. 2008. Version Control with Subversion. [En línea] 1 de Febrero de 2008. [Citado el: 10 de Diciembre de 2013.] <http://svnbook.red-bean.com/en/1.5/>.

CSS, MANUAL DE. 2012. Introducción a CSS – Manual de CSS. [En línea] 2012. [Citado el: 16 de Noviembre de 2013.] <http://www.manualdecss.com/manualdecss/introduccion-a-css/>.

EGUILUZ, JAVIER. 2011. *Desarrollo web ágil con Symfony2. 1ra edición*. 2011.

—. 2007. Introducción a JavaScript. *LibrosWeb*. [En línea] Agosto de 2007. [Citado el: 4 de Noviembre de 2013.] http://librosweb.es/javascript/capitulo_1.html.

HERNÁNDEZ ORALLO, ENRIQUE. 2013. *El Lenguaje Unificado de Modelado (UML)*. Valencia, España : s.n., 2013.

ING. MAZA OVAL, DORIS Y ING. PUPO LEYVA, ILIANNIS. 2014. *Documento de No Conformidades v1.0. Subsistema Económico. Módulo Ejecutivo*. Habana : Sistema de Informatización de los Tribunales Populares Cubanos, 2014.

ING. ROQUE HERNÁNDEZ, YOSLENYS. 2013. *Especificación de Requisitos no Funcionales de Software*. SITPC. La Habana : Universidad de las Ciencias Informáticas, 2013.

LARMAN, CRAIG. 1999. *UML y Patrones. Introducción al análisis y diseño orientado a objetos*. México : Prentice Hall, 1999. ISBN: 970-17-0261-1.

NETBEANS.ORG. 2000. Sitio oficial de NetBeans. [En línea] Junio de 2000. [Citado el: 14 de Enero de 2014.] <https://netbeans.org/about/index.html>.

POSTGRESQL, EQUIPO DESARROLLO. 2010. *Tutorial de PostgreSQL*. s.l. : Thomas Lockhart, 2010.

STEPHEN, H. KAN. 2002. *Metrics and Models in Software Quality Engineering, Second Edition*. s.l.: Addison Wesley, 2002. ISBN: 0-201-72915-6 .

SYMFONY.ES. 2009. Inyección de dependencias en Symfony 2.1.0. [En línea] 2 de Abril de 2009. [Citado el: 3 de Noviembre de 2013.] <http://www.symfony.es/noticias/2009/04/02/inyeccion-de-dependencias-en-symfony-2>.

UCI. 2012. Portal de la Universidad de las Ciencias Informáticas. *Misión*. [En línea] UCI, 2012. [Citado el: 13 de Diciembre de 2013.] <http://www.uci.cu/?q=mision>.

W3C NIGHTLY. 2013. Introduction — HTML 5.1. HTML 5.1 Nightly A vocabulary and associated APIs for HTML and XHTML. [En línea] 2013. [Citado el: 5 de Noviembre de 2013.] <http://www.w3.org/html/wg/drafts/html/master/introduction.html#html-vs-xhtml>.