



**Universidad de las Ciencias Informáticas**  
**Facultad 1**

Trabajo de Diploma para optar por el título de  
Ingeniero en Ciencias Informáticas

**Título:**

**Sistema de Gestión de la Información para el control de Plan y  
Presupuesto de la Universidad de las Ciencias Informáticas  
(SIGEPLAN).**

**Autor(es):** Jorge Pozo Hechavarria  
Rolando Ortiz Sánchez

**Tutor(es):** Msc. Suzette López Rodríguez  
MSc. Radel Calzada Pando  
Ing. Raciél Roche Escobar

**La Habana, junio de 2015**



*“Pero la juventud tiene que crear. Una juventud que no crea es una anomalía realmente. Quien no añade nada nuevo a sus conocimientos, los disminuye.”*

*Ernesto Guevara de la Serna*

## DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmamos la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

\_\_\_\_\_  
Jorge Pozo Hechavarria  
Autor

\_\_\_\_\_  
Rolando Ortiz Sánchez  
Autor

\_\_\_\_\_  
MSc. Radel Calzada Pando  
Tutor

\_\_\_\_\_  
Ing. Raciél Roche Escobar  
Tutor

\_\_\_\_\_  
Msc. Suzette López Rodríguez  
Tutora

## **DATOS DE CONTACTO**

**Msc.** Suzette López Rodríguez

Universidad de las Ciencias Informáticas, La Habana, Cuba.

Email: [suzette@uci.cu](mailto:suzette@uci.cu)

**MSc.** Radel Calzada Pando

Universidad de las Ciencias Informáticas, La Habana, Cuba.

Email: [rcalzada@uci.cu](mailto:rcalzada@uci.cu)

**Ing.** Raciél Roche Escobar

Universidad de las Ciencias Informáticas, La Habana, Cuba.

Email: [roche@uci.cu](mailto:roche@uci.cu)

## AGRADECIMIENTOS

*A mis padres que son mi orgullo y ejemplo, por estar presentes en todas las decisiones de mi vida, todo lo que soy es fruto de su amor, este título también es de ellos.*

*A mi hermanita que es una fuente de cariño en mi vida.*

*A mis segundos padres Mabel y René que las puertas de su casa siempre estuvieron abiertas para mí y me trataron como un hijo.*

*A mis abuelos que son la principal fuente de alegría, de ellos he aprendido mucho.*

*A mis tíos por su apoyo cuando decidí tomar esta universidad.*

*A mi novia por todo su cariño y dedicación, por haber compartido conmigo los últimos 3 años, con su apoyo logré terminar la carrera.*

*A mis suegros por acogerme en su familia, les debo mucho, gracias por sus sabios consejos.*

*A mis tutores, oponente y miembros del tribunal gracias a ellos se hizo posible este día realidad.*

*A mis compañeros de grupo que juntos hemos compartido 5 años de nuestras vidas.*

*A mis amigos Dorta, Escofet, Gianni, Yasiel, Arzuaga y Rolo por ser como hermanos para mí, por estar presentes en las buenas y en las malas, los llevo siempre conmigo.*

*A los colegas del 126 que han sido una familia gigante.*

*A los profesores que de una forma u otra han colaborado en mi formación profesional.*

*Gracias a los que están hoy presentes.*

*De Jorge Pozo*

*Primero que todo a mis padres, se lo que significa para ustedes este momento, y quiero agradecerles por todo el esfuerzo realizado, todos los consejos que me dieron, toda la ayuda brindada, todos los viajes a la UCI, y que sepan que este título es para ustedes también, se lo merecen por ser como son conmigo, los quiero mucho y ustedes lo saben.*

*A mi hermanita querida, la mejor del mundo, a ti también debo mucho de este logro, y aunque a veces nos fajemos, no importa, gracias y te quiero mucho.*

*A mi novia, gracias por apoyarme en todo momento, por ser como eres conmigo y por la paciencia que me tuviste todo este tiempo, te quiero mucho y te agradezco por estar siempre a mi lado.*

*A mis tutores, que sin ellos se nos hubiera dificultado mucho el resultado obtenido, gracias por toda la ayuda y atención brindada.*

*A las muchachitas del departamento de Planificación y Estadísticas, por tener para nosotros aunque fuera un minuto cada vez que las íbamos a ver, por muy atareadas que estuvieran.*

*A mi grupo, que son mi segunda familia, gracias por todos los momentos que pasamos juntos, ya fueran buenos o malos, los quiero y me quedo corto.*

*A mis compañeros de edificio, con ustedes me graduó de profesional, amigo, compañero y hermano, gracias por todos los consejos brindados y los ratos que pasamos juntos que fueron bastantes.*

*A mi compañero de tesis, por aguantarme todos estos meses, sé que no fue fácil, pero bueno aquí estamos, lo logramos y al fin somos ingenieros.*

*Por último, y no menos importante para mí, a mi familia, todos y sin excepción de ninguno, que siempre estaban arriba de mí preguntándome, ¿que como iba todo?, ¿que sí que me falta?, ¿Qué cuando terminaba?, bueno pues ya termine y se los debo a ustedes, gracias por todo, y los quiero mucho.*

*De Rolando Ortiz*

## DEDICATORIA

*A mi mamá y papá por ser los ejemplos de mi vida, los quiero mucho.*

***Jorge Pozo***

*Quiero dedicar esta tesis a todas las personas que me apoyaron en todo momento, pero sobre todo y muy especialmente a mis padres, familia y novia, que sin la ayuda de ellos no hubiera logrado ser ingeniero, muchas gracias y este logro es para ustedes también.*

***Rolando Ortiz***

## **RESUMEN**

Los Sistemas de Gestión de la Información, cuyo desarrollo está marcado por el avance de las Tecnologías de la Información y las Comunicaciones (TIC), han propiciado la optimización de los procesos en las organizaciones, lo que contribuye al control efectivo de su información.

En el área de Planificación y Estadística de la Universidad de las Ciencias Informáticas (UCI), se lleva a cabo la Elaboración del Plan y Presupuesto de dicha entidad. Este proceso es realizado de forma manual y sistemática, provocando la falta de inmediatez requerida a la hora de dar un reporte, dificultad en el control del Presupuesto e inconformidad ante los Planes que se generan por las áreas, en cuanto a las informaciones que estos deben tener presentes. En la presente investigación se desarrolla un Sistema de Gestión de la Información que permite controlar adecuadamente la información relacionada con este proceso.

Con el fin de alcanzar los objetivos trazados se analizan los principales conceptos asociados a la Elaboración del Plan y el Presupuesto, se realiza una investigación a los sistemas similares existentes, tanto en el ámbito nacional como internacional. Se expone la fundamentación de las herramientas utilizadas y los elementos necesarios a tener en cuenta para la elaboración del Sistema de Gestión de la Información para el control del Plan y Presupuesto en la UCI. Se realizan un conjunto de pruebas para garantizar el correcto funcionamiento de las funcionalidades propuestas, teniendo en cuenta también que la aplicación cumple con los requerimientos del cliente.

Como resultado se obtuvo un sistema para gestionar toda la información referente al Plan y Presupuesto, que permite además la generación de reportes y ayuda a la toma de decisiones.

**Palabras claves:** Presupuesto, Plan, Sistema de Gestión de la Información.

## ÍNDICE DE CONTENIDO

|   |    |
|---|----|
| INTRODUCCIÓN.....   | 1  |
| Capítulo I: FUNDAMENTACIÓN TEÓRICA.....   | 6  |
| 1.1.    Introducción.....   | 6  |
| 1.2.    Principales conceptos asociados a la gestión del Plan y Presupuesto económico.....                | 6  |
| 1.3.    Sistemas para la gestión del Plan y Presupuesto, utilizados a nivel nacional e internacional..... | 8  |
| 1.3.1.  Sistemas en el ámbito Internacional.....  | 9  |
| 1.3.2.  Sistemas en el ámbito nacional.....   | 10 |
| 1.4.    Metodologías de desarrollo de software.....   | 13 |
| 1.4.1.  Metodologías tradicionales.....   | 14 |
| 1.4.2.  Metodologías ágiles.....  | 14 |
| 1.5.    Herramientas y Tecnologías.....   | 18 |
| 1.5.1.  Tecnologías de programación del lado del servidor.....  | 18 |
| 1.5.2.  Lenguaje de modelado.....   | 18 |
| 1.5.3. <i>Framework</i> de Interfaz.....  | 19 |
| 1.5.4. <i>Framework</i> de desarrollo.....  | 20 |
| 1.5.5.  Servidor web.....   | 20 |
| 1.5.6.  Herramienta CASE.....   | 21 |
| 1.5.7.  IDE de desarrollo.....  | 22 |
| 1.5.8.  Sistema Gestor de Base de Datos (SGBD).....   | 22 |
| 1.5.9.  Notación de Modelado de Procesos de Negocio (BPMN).....   | 22 |
| 1.6.    Conclusiones Parciales.....   | 23 |
| CAPÍTULO II: ANÁLISIS Y DISEÑO DEL SISTEMA DE GESTIÓN DE LA INFORMACIÓN DE PLAN Y PRESUPUESTO.....        | 24 |
| 2.1.    Introducción.....   | 24 |
| 2.2.    Descripción general de la propuesta de solución.....  | 24 |
| 2.3.    Modelado de procesos de negocio.....  | 25 |
| 2.3.1.  Subproceso: Llenar el modelo de las partidas correspondientes a cada área.....                    | 26 |
| 2.3.2.  Subproceso: Emitir el modelo del Plan por área.....   | 27 |
| 2.3.3.  Subproceso: Conformar los datos del modelo del Plan por partidas asociadas a cada área.....       | 28 |

|   |    |
|---|----|
| 2.3.4. Roles que intervienen en el proceso de negocio. ....   | 29 |
| 2.4. Requisitos del sistema. ....   | 29 |
| 2.4.1. Requisitos funcionales (RF). ....  | 30 |
| 2.4.2. Requisitos no funcionales (RNF). ....  | 30 |
| 2.5. Planificación. ....  | 32 |
| 2.5.1. Historia de usuario (HU). ....   | 32 |
| 2.5.2. Plan de entregas. ....   | 34 |
| 2.5.3. Plan de iteraciones. ....  | 35 |
| 2.6. Diseño. ....   | 36 |
| 2.6.1. Descripción de la arquitectura. ....   | 36 |
| 2.6.2. Patrones de diseño. ....   | 38 |
| 2.6.3. Diseño de la base de datos. ....   | 40 |
| 2.7. Conclusiones Parciales. ....   | 41 |
| CAPÍTULO III: IMPLEMENTACIÓN Y PRUEBAS DEL SISTEMA DE GESTIÓN DE LA INFORMACIÓN DE PLAN Y PRESUPUESTO. .... | 42 |
| 3.1. Introducción. ....   | 42 |
| 3.2. Estándares de codificación. ....   | 42 |
| 3.3. Implementación. ....   | 44 |
| 3.3.1. Diagrama de despliegue. ....   | 44 |
| 3.4. Pantallas principales de la aplicación. ....   | 46 |
| 3.5. Validación de la solución propuesta. ....  | 48 |
| 3.5.1. Pruebas unitarias. ....  | 48 |
| 3.5.2. Pruebas de aceptación. ....  | 52 |
| 3.5.3. Validación de las variables de la investigación. ....  | 54 |
| 3.6. Conclusiones Parciales. ....   | 56 |
| CONCLUSIONES GENERALES. ....  | 57 |
| RECOMENDACIONES. ....   | 58 |
| BIBLIOGRAFÍA. ....  | 59 |
| GLOSARIO DE TÉRMINOS. ....  | 63 |

## INTRODUCCIÓN

La planificación presupuestaria es un proceso muy técnico, cuyo propósito es traducir las prioridades y los proyectos del Plan anual de operaciones en un cronograma de financiación real, con descripciones detalladas de requerimientos y compromisos de financiación de proyectos (Colina Luisa, 2012).

El gobierno cubano se encuentra en un proceso de informatización de todos los sectores de la economía. La UCI se encuentra inmersa dentro de estos cambios tecnológicos, es una entidad presupuestada perteneciente al Ministerio de Educación Superior (MES), la cual pretende servir de soporte a la industria cubana del software. La UCI en este sentido puede apoyar a los procesos económicos en instituciones del país que lo requieran, así como en su infraestructura empresarial. En la Vice-Rectoría Económica se encuentra el área de Dirección de Planificación y Estadística, la cual presenta como misión llevar a cabo la Elaboración del Plan y Presupuesto en la Universidad.

La UCI cuenta con un Plan y un Presupuesto anual de gastos corrientes dividido en peso cubano convertible (CUC) y peso cubano (CUP). Estos se desagregan en los 7 elementos de gastos autorizados en el país, 27 sub-elementos y 150 partidas para tener un mejor control y análisis de sus ejecuciones. Todas estas partidas se encuentran distribuidas en diferentes áreas de las cuales 18 son administradores de partidas, los cuales proponen y controlan de forma primaria el Plan y Presupuesto, el resto de las áreas son consumidoras. Además existe un Plan y un Presupuesto para inversiones, lo anterior se resume en 32 modelos para el Plan y 24 modelos para el Presupuesto, desagregados por sus diferentes categorías respectivamente.

La UCI utiliza la herramienta Sistema de Control Presupuestario y Financiero en el área de Planificación y Estadística la cual se encarga de gestionar la ejecución financiera del Presupuesto. Dentro de la ejecución financiera, se tienen pagos en servicios y materiales. Este sistema contable ejecuta este proceso excepto la Elaboración del Plan y Presupuesto. Esta aplicación carece de una funcionalidad para realizar dicha gestión trayendo como resultado que este proceso se realice manualmente, auxiliándose de tablas en Microsoft Excel, surgiendo los siguientes inconvenientes:

- El mecanismo de control de la Elaboración del Plan y Presupuesto no es eficiente, provocando la falta de inmediatez requerida a la hora de dar un reporte.

- Se tiene dificultad en el control del Presupuesto, debido a que no se tiene conocimiento preciso de los Presupuestos que se generan por cada área del centro.
- Inconformidad ante los Planes que se generan por las áreas en cuanto a las informaciones que estos deben tener presentes, para llegar a ejecutar durante todo un año.

La Dirección tiene la responsabilidad de controlar y darle seguimiento a todas las peticiones realizadas por las diferentes áreas para que no ocurran sobregiros o inejecuciones de las partidas.

Por las razones anteriormente expuestas queda planteado el siguiente **problema de la investigación**: ¿Cómo facilitar la gestión de la información referente al Plan y Presupuesto de la UCI de manera que se garantice el control y agilice el procesamiento de la información?

Queda definido como **objeto de estudio** de la presente investigación: los sistemas de gestión de Plan y Presupuesto económico.

Para darle solución al problema antes descrito se tiene como **objetivo general**: Desarrollar un Sistema de Gestión de la Información referente al Plan y Presupuesto de la UCI que garantice el control y agilice el procesamiento de la información.

El **campo de acción** se enmarca en los sistemas para el control de la propuesta de Plan y Presupuesto.

Para darle solución al **objetivo general** se definen los siguientes **objetivos específicos**:

- 1) Elaborar el marco teórico de la investigación mediante un estudio del estado del arte sobre el desarrollo de aplicaciones para la gestión de la información de Plan y Presupuesto económico.
- 2) Obtener los requisitos funcionales y no funcionales del sistema.
- 3) Diseñar el Sistema de Gestión de la Información referente al Plan y Presupuesto.
- 4) Implementar el Sistema de Gestión de la Información referente al Plan y Presupuesto.
- 5) Validar la propuesta de solución mediante pruebas de software y el cliente software.

La **idea a defender** queda definida en el siguiente planteamiento: Con la implementación de una herramienta que facilite la gestión de la información del Plan y Presupuesto, se garantiza el control y se agiliza el procesamiento de la información en la UCI.

Las **tareas de la investigación** para dar cumplimiento a los objetivos planteados son:

- 1) Revisión de documentos y entrevista a directivos, jefes de partidas, trabajadores de la Dirección de Planificación y Estadística, así como de la Dirección de Contabilidad y Finanzas y Vicerrectoría Económica.
- 2) Caracterización de los sistemas en el ámbito nacional e internacional que realizan la gestión del Plan y Presupuesto.
- 3) Selección y caracterización de los métodos, herramientas y tecnologías para el desarrollo del sistema.
- 4) Obtención de los requisitos funcionales y no funcionales del sistema.
- 5) Implementación de un Sistema de Gestión de la Información referente al Plan y Presupuesto.
- 6) Elaboración y aplicación de los casos de prueba para la propuesta de solución.

Se emplean los métodos científicos de la investigación: Teóricos y Empíricos. De los métodos teóricos se utiliza:

- **Analítico-Sintético:** La aplicación de este método permitió la búsqueda, investigación y análisis de los documentos necesarios para entender el proceso que se lleva a cabo en la elaboración del Plan y Presupuesto anual. Además se consultaron disímiles fuentes bibliográficas.
- **Análisis Histórico-Lógico:** Se utiliza para conocer los antecedentes y elementos de la investigación referidos a los procesos de planificación y control y la determinación de las tendencias actuales sobre la temática que se estudia (Hernández, y otros, 2011). Su empleo permitió investigar acerca de las herramientas informáticas existentes en Cuba y en el mundo que gestionan Planes y Presupuestos.

- **Modelación:** Es el método mediante el cual se crean abstracciones con el objetivo de explicar la realidad (Hernández, y otros, 2011). Se empleó para la confección de los prototipos funcionales, la representación de los requisitos del sistema y el diseño de la base de datos.

De los métodos empíricos se utiliza:

- **Entrevista:** Se realizó en las áreas, con sus respectivos especialistas, las cuales se ven inmersas en el proceso de la elaboración del Plan y Presupuesto, así como las áreas y personas que no están involucradas en la elaboración, pero manejan información referente al tema, el objetivo de estas entrevistas es elaborar una aplicación libre de errores y obtener las recomendaciones necesarias para el desarrollo de la misma.

Los **aportes prácticos esperados** con la elaboración de la aplicación son los siguientes:

- Realizar una aplicación que se encargue de gestionar la información del Plan y Presupuesto en la Universidad.
- Proporcionar la información del Plan y Presupuesto para cada una de las áreas pertinentes, con sus respectivos permisos de accesibilidad.
- Permitir la seguridad de las informaciones que se generen por el Plan y Presupuesto.

La estructura del trabajo de diploma está compuesta por tres capítulos, resumen, introducción, conclusiones, recomendaciones, referencias bibliográficas, anexos y un glosario de términos. A continuación, se describen los principales aspectos abordados en cada uno de los capítulos:

### **Capítulo I: Fundamentación teórica.**

En este capítulo se puntualizan los principales conceptos relacionados con el tema, se realiza el estudio del estado del arte a nivel nacional e internacional sobre los Sistemas de Gestión de la Información de Presupuestos y Planes. Además se definen las herramientas, metodología y tecnologías a utilizar durante el desarrollo del sistema.

### **Capítulo II: Análisis y Diseño del Sistema de Gestión de la Información de Plan y Presupuesto.**

En este capítulo se expone una descripción general de la solución propuesta y su funcionamiento, se capturan y se validan los requisitos funcionales y no funcionales y otros elementos importantes dentro del proceso de desarrollo, tales como: arquitectura, diseño del sistema, patrones empleados y los artefactos derivados de la metodología de desarrollo de software seleccionada.

### **Capítulo III: Implementación y Pruebas del Sistema de Gestión de la Información de Plan y Presupuesto.**

En este capítulo se abordan aspectos relacionados con la implementación del sistema en base a la arquitectura de desarrollo de software. Se documentan las pruebas de software al Sistema de Gestión de la Información de Plan y Presupuesto, realizándose las pruebas unitarias y de aceptación para verificar que responda a un correcto funcionamiento, garantizar la confidencialidad y detectar fallas en el sistema.

## **Capítulo I: FUNDAMENTACIÓN TEÓRICA.**

### **1.1. Introducción.**

En este capítulo se abordan diferentes conceptos para entender los procesos de Elaboración del Plan y Presupuesto. Se realiza un estudio de los sistemas existentes que estén estrechamente relacionados con la Elaboración del Plan y Presupuesto económico de una entidad en el ámbito nacional e internacional. Se hace un análisis de las metodologías de desarrollo de software, tecnologías, y herramientas actuales, de manera tal, que se seleccionen las más convenientes para la realización del sistema a implementar.

### **1.2. Principales conceptos asociados a la gestión del Plan y Presupuesto económico.**

Un Plan es una ficha clave en la gestión empresarial, pues permite orientar lo que se va a realizar en el futuro de acuerdo a las necesidades de una empresa o institución. Por consiguiente un plan precisa los detalles necesarios para realizar una obra. Tal es su importancia que muchos autores han conceptualizado sobre el tema.

- En su forma más simple, el concepto de Plan se define como la intención y proyecto de hacer algo, o como proyecto que, a partir del conocimiento de las magnitudes de una economía, pretende establecer determinados objetivos. Asimismo se ha definido como un documento en que se constan las cosas que se pretenden hacer y la forma en que se piensa llevarlas a cabo (Ordaz Zubia, y otros, 2006).
- Un Plan es una descripción de las intenciones amplias y a corto plazo, en cuanto a lo que uno ve que se requiere para ocuparse de un área específica. Se espera que un plan remedie circunstancias no óptimas en un área o que la expanda o que obstruya o impida una oposición a la expansión. Para que un plan se lleve a cabo se requiere que se desglose en las acciones específicas que son necesarias para lograr lo que el plan se propone hacer (Iglesia de Scientology Internacional, 2011).

Después de haber estudiado los conceptos definidos por otros autores, los autores del presente trabajo deciden tomar como concepto de Plan el que expone Ordaz Zubia porque se ajusta más al proceso que se realiza en la UCI.

El Presupuesto como instrumento clave de la gestión empresarial, llega a ser tan significativo que si en una organización estos no funcionan, se debe replantear la forma en que se realizan. Por su importancia varios autores han conceptualizado sobre el tema.

- Un Presupuesto es un Plan integrador que se expresa en términos financieros con respecto a las operaciones y recursos que forma parte de una empresa para un período determinado, con el fin de lograr los objetivos fijados por la gerencia general (Burbani, 2005).
- El Presupuesto es un Plan de acción dirigido a cumplir una meta prevista, expresada en valores y términos financieros que, debe cumplirse en determinado tiempo y bajo ciertas condiciones previstas (Leiseca, 2010).

Luego de haber estudiado los conceptos definidos por otros autores, referentes al Presupuesto, los autores de la presente investigación toman el concepto que expone Leiseca, porque se ajusta más a la investigación y a los procesos que se pretenden informatizar.

Las partidas de gastos encargadas de recoger los gastos de una entidad, están diseñadas para llevar el control de lo que se desea aplicar. Varios autores han conceptualizado sobre estas partidas.

- Las partidas resumen los diferentes conceptos primarios de gastos y son agrupaciones que están en correlación con las definidas en los lineamientos para la determinación de los costos (MFP, 2014).
- La partida es un concepto económico asociado al costo de producción, empleado para agrupar los gastos, identificar el lugar donde éstos se originan y la forma directa o indirecta en que inciden en el costo. Se utiliza en la determinación de los costos, tanto en la etapa del plan como en el real, pudiendo crearse para ello la nomenclatura necesaria para identificarlas según los requerimientos del proceso productivo de cada empresa. El uso de la partida permite la precisión e información del comportamiento de los costos, creando las premisas que facilitan el control oportuno de las desviaciones surgidas durante la ejecución del Plan (CASA CONSULTORA DISAIC, 2015).

Teniendo en cuenta lo anteriormente planteado, los autores de la presente investigación toman el concepto que expone el Ministerio de Finanzas y Precios (MFP) porque es el concepto por el cual se rige el área de Planificación y Estadísticas.

Los Sistemas de Gestión de la Información encargados de la seguridad y protección de la información que manejan, están diseñados con el propósito de proteger los activos informáticos en una entidad. Algunos autores han conceptualizado sobre estos sistemas.

- Proceso mediatizado por un conjunto de actividades que permiten la obtención de información, lo más pertinente, relevante y económica posible, para ser usada en el desarrollo y el éxito de una organización. Genera nuevos conocimientos. Un Sistema de Gestión de Información permite la gestión de los recursos de información tanto internos como externos. Su finalidad es generar servicios y productos que respondan a las necesidades y sobrepasen las expectativas de los usuarios, posibilitando que el sistema trabaje eficientemente y económicamente a la vez. El Sistema de Gestión de Información aprovecha al máximo sus recursos de información en función de la mejora continua y de la toma de decisiones organizacional a todos los niveles jerárquicos desde la cúspide estratégica hasta la base operativa (Ecured, 2015).
- Sirven para el registro de las transacciones diarias y la generación de reportes que presentan información. relevancia, claridad, sencillez y oportunidad de tal forma que sea útil para las personas a quienes se les entrega. Se ve su uso en muchas empresas, que van desde una clasificación de micro hasta gran empresa; sin embargo, la aplicación en cada una puede variar debido a la magnitud de actividades de la misma, no así por su tamaño (Scribd. Inc, 2015).

Luego de haber analizado los conceptos definidos por otros autores, los autores del presente trabajo definen como concepto el siguiente:

**Sistema de Gestión de la Información:** Es un sistema donde el usuario tenga una constante interacción con los datos relacionados. El sistema debe ser capaz de propiciar la seguridad de la información y asegurar la integridad y confidencialidad de los activos de la información.

### **1.3. Sistemas para la gestión del Plan y Presupuesto, utilizados a nivel nacional e internacional.**

A continuación se expone una breve descripción de algunos Sistemas de Gestión de la Información referente al Plan y Presupuesto con el propósito de arribar a conclusiones que permitan guiar la propuesta de solución hacia el cumplimiento de los objetivos propuestos.

### **1.3.1. Sistemas en el ámbito Internacional.**

**SIPPRES:** La Oficina de Planificación Universitaria de la Universidad de Costa Rica (OPLAU), puso en funcionamiento el nuevo Sistema Institucional para la Formulación de Proyectos y Plan-Presupuesto (SIPPRES). Esta plataforma permite la captura de información para la inscripción, aprobación y puesta en marcha de los proyectos académicos en la Universidad de Costa Rica. Los investigadores tendrán la posibilidad de inscribir proyectos inter-viceerectorías, lo que les permite formular proyectos con varias actividades sustantivas, así como solicitar recursos a cada Viceerectoría, según corresponda. Esta nueva plataforma permite además desglosar actividades y tareas que se desarrollan sacando el cálculo por horas y por tareas. Todo esto facilitará contar con datos y herramientas para un uso más eficiente de los recursos institucionales (Parral, 2014).

Esta herramienta a pesar de contar con funcionalidades que gestionan el Plan y Presupuesto no se puede implantar en la Universidad. No se ajusta a la manera en que se gestiona el Plan y Presupuesto en la UCI, lo cual es de vital importancia para el sistema a implementar. Además se desconoce el código fuente.

**Softland ERP:** “Es una solución de software empresarial diseñada específicamente para cumplir con los requerimientos de la empresa mediana en Latinoamérica. Es un sistema de gestión totalmente integrado y escalable que brinda una integración total con soluciones en el área de Inteligencia de Negocios, CRM (por sus siglas en inglés *Customer Relationship Management*), Aplicaciones Web (Portal de Personal y Comercio Electrónico) y Aplicaciones Móviles (Facturación en Ruta y Administración de Almacén)” (Softland Inversiones S.L., 2015). Algunos de los módulos que componen este sistema son: Contabilidad General, Presupuesto Financiero, Activos Fijos, Control Presupuestal, Recursos Humanos & Nómina y Ventas e Inventario.

El módulo de Presupuesto Financiero facilita las comparaciones entre los presupuestos aprobados y los resultados reales extraídos de la contabilidad general. Conjuntamente, el módulo de Control Presupuestal permite el control de los ingresos y egresos provocados por las transacciones que afecten las diferentes

áreas de la organización y establecer comparaciones entre los presupuestos. Brinda además, un manejo efectivo de la seguridad de la información del sistema pues deja constancia de modificaciones presupuestarias y/o justificaciones (Softland Inversiones S.L., 2015).

La tecnología que se utilizó para el desarrollo del sistema Softland ERP está basada en software propietario y es muy difícil para Cuba adquirir el software y pagar las licencias y sus actualizaciones.

**Presto:** Es una herramienta internacional desplegada por la empresa desarrolladora de productos informáticos Soft. Es la más utilizada para realizar Presupuestos de edificación, obra civil y empresas constructoras. Entre sus atractivos se encuentra que: (Jesús Granda, y otros, 2007)

- Permite realizar fácilmente Presupuestos y ofertas reutilizando información de proyectos anteriores o bases de datos de precios.
- Edición simultanea de varios Presupuestos o bases de datos de precios.
- Uso de arrastrar y soltar, copiar y pegar para traspasar información entre ventanas, entre Presupuestos o con otros programas, como hojas de cálculo y editores de textos.
- Utilidades para copiar, pegar, renombrar, mover, eliminar, comprimir, descomprimir y enviar un presupuesto por correo electrónico sin abandonar Presto.

Esta herramienta a pesar de contar con funcionalidades presupuestales, no se ajusta a los requisitos necesarios para resolver el problema de la planificación y el control del Presupuesto en el área de Planificación y Estadística de la UCI. Primeramente porque está diseñada para efectuar Presupuestos de edificación y obra civil, lo cual está muy distante del objetivo que se persigue con la elaboración del presente trabajo. Sus requisitos de instalación no cumplen con la política del país de migrar a software libre, esto hace que se incurra en un gasto elevado en pagos de licencia de software.

### **1.3.2. Sistemas en el ámbito nacional**

**Versat-Sarasola:** Sistema cubano de contabilidad confiable, permite enviar información eficaz, de forma inmediata, desde lugares apartados, a la vez que ofrece mayor organización, control y disciplina en cada gestión. Constituye un software que automatiza prácticamente todas las actividades de Planificación, Control y Análisis Económico de cualquier tipo de Entidad en el país o fuera del mismo (Martinez, 2015).

Versat-Sarasola está formado por 12 módulos o subsistemas, entre los cuales se encuentra el módulo de Finanzas, Caja y Banco. Este es el subsistema más abarcador del Versat-Sarasola pues está conformado por cinco actividades: Caja, Banco, Cobros y Pagos, Crédito Bancario y Otras Operaciones Financieras. Es en esta última actividad que se encuentra el manejo de los Presupuestos. Sin embargo las funcionalidades que realiza no se necesitan para planificar y controlar el Presupuesto en la Dirección de Planificación y Estadísticas. Dichas funcionalidades se mencionan a continuación:

- Planificar los pagos a efectuar que generan salida de efectivo.
- Planificar los cobros a efectuar que generan entrada de efectivos.
- Se realiza un documento que recoge los aportes del Presupuesto del Estado, como por ejemplo, el pago de la seguridad social.(VERSAT, 2012)

Además, en el módulo Costos y Procesos se realizan la actividad Elementos de Gastos, en donde se definen los elementos de gastos que se utilizarán en el sistema. Para lo cual se encuentran las opciones Adicionar elemento y Eliminar elemento, pero al ejecutar esta última se borran todas las áreas asociadas (VERSAT, 2012). Considerándose de esta manera, que este subsistema cumple parcialmente con uno de los requisitos presentes en la herramienta a implementar. Por la información antes mencionada, se considera que este sistema no se corresponde con las necesidades presentadas por la Dirección de Planificación y Estadística para el desarrollo de la herramienta que permita la planificación y control del Presupuesto.

**Sistema de Control Presupuestario y Financiero (SCPF):** Sistema informático desarrollado en la UCI para controlar el flujo de trabajo relacionado con las solicitudes de compra y ejecución del presupuesto de la Vice-Rectoría Económica. Para llevarlo a cabo se utilizó como lenguaje de programación C#.NET y como gestor de base de datos SQL Server 2000. El mismo consta de dos módulos, el primero recoge todo el flujo de trabajo que se realiza en el área de Planificación y Estadística, y el segundo recoge todo el proceso que se realiza en el área de Finanzas. Su principal objetivo es la informatización de todos los procesos que realiza la Vice-Rectoría Económica y dentro de ellos se determinó con mayor prioridad el control presupuestario y financiero de las solicitudes que se presentan en el comité de compras. A continuación se presentan las funcionalidades que el sistema proporciona:

- Inserción de solicitudes al sistema alertando cuando no existe Presupuesto para procesar dicha solicitud.
- Adiciona cuentas bancarias así como el Presupuesto para cada una de ellas y el saldo inicial en banco con que inician el año.
- Adiciona todos los elementos de gastos.
- Adiciona todos los especialistas que participan en el comité de compras.
- Gestiona el estado de la solicitud (enviada al rector, no aprobada, aprobada, enviada a finanzas).
- Inserta cheques asociándolos a la solicitud correspondiente.
- Presenta reportes para facilitar el análisis y procesamiento de la información existente en el sistema.

La herramienta efectúa el control de Presupuestos, pero no lo planifica, además está desarrollada en tecnologías privativas, lo cual es política de la Universidad migrar a tecnologías libres. Estas razones fortalecen para no emplearla en el sistema a implementar. Cabe destacar que se tomaron algunas funcionalidades como Adicionar todos los especialistas que participan en el comité de compras y Gestionar el estado de la solicitud con el fin de analizar su implementación.

**eTES ERP:** “Es un sistema de planificación de recursos empresariales (ERP), que tiene como propósito fundamental apoyar la gestión empresarial, otorgando tiempos rápidos de respuesta a los problemas, así como un eficiente manejo de información que permita la toma oportuna de decisiones y disminución de los costos totales de operación”. Desarrollado en Cuba y diseñado sobre una arquitectura multicapas. Está compuesto por módulos interconectados que gestionan los procesos de un negocio: Compras, Inventario, Producción, Ventas, Contabilidad y Finanzas, Recursos Humanos, Activos Fijos y Consolidación de Información (eTES, 2015).

Este sistema no tiene ningún módulo que controle la ejecución del Plan y Presupuesto en las entidades cubanas, aunque debe tenerse en cuenta que el módulo de Consolidación de Información genera reportes de los resultados obtenidos.

Se muestra a continuación un cuadro comparativo de los sistemas estudiados:

| <b>Características/Sistemas</b>                    | <b>SIPPRES</b> | <b>Softland ERP</b> | <b>Presto</b> | <b>Versat-Sarasola</b> | <b>SCPF</b> | <b>eTE S ERP</b> |
|--|----------------|---------------------|---------------|------------------------|-------------|------------------|
| Control de la ejecución del Plan.                  | sí             | sí                  | sí            | no                     | no          | no               |
| Desarrollado con tecnologías libres.               | no             | no                  | no            | no                     | no          | no               |
| Manejo efectivo de la seguridad de la información. | sí             | sí                  | sí            | sí                     | sí          | sí               |
| Generación de reportes.                            | sí             | sí                  | sí            | sí                     | sí          | sí               |

**Tabla 1: Análisis de los sistemas estudiados.**

Los módulos que componen las aplicaciones desarrolladas en Cuba no son suficientes para cubrir en su totalidad el proceso de negocio Elaboración del Plan y Presupuesto que se desarrolla en el área de Planificación y Estadística de la UCI; los sistemas extranjeros controlan la ejecución del Plan, pero con sus propias particularidades las cuales no se ajustan a la manera en que realiza el proceso en la UCI, además tienen el inconveniente de ser privativos y para poder utilizar alguno de sus módulos se deberían pagar la licencia, lo que resultaría costoso para la UCI. Analizando todos estos elementos se hace inminente la necesidad de implementar un Sistema de Gestión de la Información para el control del Plan y Presupuesto que cumpla con la totalidad de los requerimientos funcionales del proceso.

#### **1.4. Metodologías de desarrollo de software.**

Una metodología de desarrollo de software se define como un conjunto de procedimientos, técnicas, herramientas y un soporte documental que ayuda en la construcción de un software (Pressman, 2001).

Las metodologías para el desarrollo del software imponen un proceso disciplinado sobre el desarrollo de software, con el fin de hacerlo más predecible y eficiente. Además tienen como principal objetivo aumentar la calidad del software que se produce en todas y cada una de sus fases de desarrollo. El desarrollo de software, no es una tarea fácil. Prueba de ello es que existen numerosas metodologías de desarrollo, que inciden en distintas dimensiones del proceso de desarrollo.

#### **1.4.1. Metodologías tradicionales.**

Entre las metodologías propuestas para el desarrollo de software se tienen las tradicionales que se centran especialmente en el control del proceso, estableciendo rigurosamente las actividades involucradas, los artefactos que se deben producir, y las herramientas y notaciones que se usarán. Estas metodologías se usan especialmente para proyectos grandes donde el cliente interactúa con el equipo de desarrollo mediante reuniones. Las propuestas han demostrado ser efectivas y necesarias en un gran número de proyectos, pero también han presentado problemas en muchos otros. Ante las dificultades de restricciones de tiempo y flexibilidad que poseen estas metodologías, otras más actuales han sido adoptadas para guiar el proceso de desarrollo, denominadas “metodologías modernas”, dentro de las cuales figuran las “metodologías ágiles” (Córdova Dorta, 2011).

#### **1.4.2. Metodologías ágiles.**

Las metodologías ágiles, por el contrario, ponen énfasis en entregar buen código al cliente. En obtener resultados que satisfagan al cliente, adaptándose a sus siempre cambiantes necesidades. Aunque esto no quiere decir que las metodologías ágiles abandonen totalmente herramientas y procedimientos ya establecidos, ni que renuncien a la disciplina. Tan solo significa que los emplean de forma flexible. Lo principal es obtener buenos resultados, no el hacer trabajos perfectos. A continuación se exponen los principales objetivos que presentan estas metodologías.

- Flexibilidad ante los cambios que quisieran efectuar los clientes a lo largo del desarrollo de un sistema.
- Depositar en manos del cliente los elementos desarrollados en cada etapa del trabajo para que este realice las verificaciones pertinentes.
- Evitar exceso de documentación y malentendidos trabajando en conjunto entre el cliente y el equipo de desarrollo con una comunicación directa (Orjuela, 2008).

Se decide utilizar en el trabajo de diploma, metodologías ágiles, ya que se adaptan más al entorno en el que se está trabajando. El equipo de desarrollo para la realización de la propuesta es pequeño, conformado por solo dos personas, y necesita que el cliente sea parte del desarrollo. Este enfoque está

mostrando su efectividad en proyectos con requisitos muy cambiantes y cuando se exige reducir drásticamente los tiempos de desarrollo, pero manteniendo una alta calidad.

A continuación se caracterizan algunas metodologías ágiles existentes:

- ***eXtreme Programming***

Programación Extrema (En inglés abreviado XP) es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico. El ciclo de desarrollo consta de cuatro fases (Oswaldo, y otros, 2014):

1. **Planificación del Proyecto.**

Se tiene las **Historias de usuario**, las cuales son usadas para estimar tiempos de desarrollo de la parte de la aplicación que describen. El **Plan de entregas**, el cual es una planificación donde los desarrolladores y clientes establecen los tiempos de implementación ideales de las historias de usuario y el **Plan de iteraciones** evidencia que al comienzo de cada iteración los clientes deben seleccionar las historias de usuario definidas en el Plan de entregas que serán implementadas. La **velocidad del proyecto** también se encuentra adjunto a esta fase, es una medida que representa la rapidez con la que se desarrolla el proyecto, la **programación en pareja**, incrementa la productividad y la calidad del software desarrollado y las **reuniones diarias** las cuales definen: es necesario que los desarrolladores se reúnan diariamente y expongan sus problemas, soluciones e ideas de forma conjunta.

2. **Diseño.**

Difunde los **diseños simples**, los cuales sugieren un diseño fácilmente entendible que a la larga costará menos tiempo y esfuerzo desarrollar. El **glosario de términos** propicia una correcta especificación de los nombres de métodos y clases, la **funcionalidad extra** se encarga de alertar a los desarrolladores que

nunca se debe añadir funcionalidad extra al programa aunque se piense que en un futuro será utilizada. La **refactorización** propone mejorar y modificar la estructura así como la codificación de códigos ya creados sin alterar su funcionalidad y por último, las **tarjetas CRC** permiten al programador centrarse y apreciar el desarrollo orientado a objetos olvidándose de los malos hábitos de la programación procedural clásica.

### **3. Codificación.**

Esta fase propone que la codificación debe hacerse atendiendo a estándares de codificación ya creados.

### **4. Pruebas.**

Aquí la metodología propone las pruebas de aceptación para comprobar el funcionamiento de los códigos, dichas pruebas son creadas y usadas por los clientes para comprobar que las distintas historias de usuario cumplen su cometido.

- **SCRUM**

Desarrollada por Ken Schwaber, Jeff Sutherland y Mike Beedle. Define un marco para la gestión de proyectos, que se ha utilizado con éxito durante los últimos 10 años. Está especialmente indicada para proyectos con un rápido cambio de requisitos. Sus principales características se pueden resumir en dos. El desarrollo de software se realiza mediante iteraciones, denominadas *sprints*, con una duración de 30 días. El resultado de cada *sprint* es un incremento ejecutable que se muestra al cliente. La segunda característica importante son las reuniones a lo largo proyecto, entre ellas destaca la reunión diaria de 15 minutos del equipo de desarrollo para coordinación e integración (Informáticas, 2013).

- **Crystal Methodologies**

Se trata de un conjunto de metodologías para el desarrollo de software caracterizadas por estar centradas en las personas que componen el equipo y la reducción al máximo del número de artefactos producidos. Han sido desarrolladas por Alistair Cockburn. El desarrollo de software se considera un juego cooperativo de invención y comunicación, limitado por los recursos a utilizar. El equipo de desarrollo es un factor clave, por lo que se deben invertir esfuerzos en mejorar sus habilidades y destrezas, así como tener políticas de

trabajo en equipo definidas. Estas políticas dependerán del tamaño del equipo, estableciéndose una clasificación por colores, por ejemplo Crystal Clear (3 a 8 miembros) y Crystal Orange (25 a 50 miembros) (Informáticas, 2013).

- **OpenUp**

Proceso Unificado Abierto (en inglés abreviado OpenUp) es una metodología planteada para equipos de desarrollo pequeños, lo que propicia obtener resultados en un corto período de tiempo. Es ágil e incremental y para proyectos de software basados en desarrollo iterativo. Plantea que se debe utilizar sólo los procesos que sean necesarios, sin demasiados artefactos y sobre todo que el proyecto debe acoplarse a las necesidades del usuario. Esta metodología tiene ventajas importantes como que disminuye los riesgos y puede utilizarse tanto en proyectos pequeños como en proyectos grandes, aunque está concebida para proyectos pequeños. Si se hace un correcto uso de esta se puede desarrollar un software de gran calidad, a pesar de que se le diseñe en poco tiempo y con poca documentación (Vázquez Chacón, 2013).

Se escogió la metodología XP para realizar el Sistema de Gestión de la Información, para el control de Plan y Presupuesto de la UCI, ya que XP, se ajusta a las características del sistema a realizar, debido a que la metodología en sus principios básicos plantea la programación en equipos pequeños, pudiendo los miembros del equipo, intercambiar responsabilidades en un momento determinado. Otro argumento se basa en la realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. Además posee las siguientes ventajas:

- Programación organizada.
- Menor tasa de errores.
- Satisfacción del programador.
- Solución de errores de programas.
- Versiones nuevas.

- Implementa una forma de trabajo donde se adapte fácilmente a las circunstancias.

## **1.5. Herramientas y Tecnologías.**

Hoy en día existe un gran avance en el sector informático, donde poco a poco van apareciendo nuevas tecnologías y herramientas con el fin de hacer más reales las soluciones de los clientes. La fusión de las tecnologías de desarrollo con las herramientas ha permitido desarrollar aplicaciones web para la gestión de la información, en la cual el usuario final está implícito en el proceso de desarrollo, para lograr un resultado más real y acorde a lo esperado.

### **1.5.1. Tecnologías de programación del lado del servidor.**

**Personal Home Page Tools (PHP):** En su versión 5.4.9 es una tecnología del lado del servidor, de código abierto y diseñado en sus inicios para el desarrollo de páginas web dinámicas. Una de sus principales características es la capacidad de soportar gran cantidad de bases de datos. También permite la integración con varias bibliotecas externas permitiéndole al programador analizar código XML y generar documentos en diferentes formatos. Este lenguaje es rápido, libre, orientado a objetos y multiplataforma, pues permite ser utilizado sobre diferentes sistemas operativos como: Linux y Windows. Posee además una amplia librería de funciones y cuenta con una extensa documentación, la cual permite un rápido aprendizaje (Doyle, 2012).

#### **Ventajas que proporciona:**

- Permite la integración con disímiles tipos de servidores de bases de datos tales como: MySQL, PostgreSQL y Oracle.
- Permite emplear técnicas de programación orientada a objetos.
- Es libre y multiplataforma.

### **1.5.2. Lenguaje de modelado.**

**Unified Modeling Language (UML):** En su versión 2.0, es el lenguaje gráfico más conocido y utilizado en la actualidad para visualizar, especificar y documentar cada una de las partes que comprende el desarrollo

de software. Este lenguaje es semejante al de la vida real, claro y uniforme para el diseño orientado a objetos, ya que permite la fuerte integración entre herramientas, procesos y dominios. UML se especializa en el modelado de elementos conceptuales como son: procesos de negocio y funciones de los sistemas. También este lenguaje permite escribir clases en un lenguaje determinado, esquemas de base de datos y componentes de software reusables (Ferreira dos Santos, 2012).

### 1.5.3. *Framework de Interfaz.*

**Bootstrap:** En su versión 3.0.1 es un entorno de trabajo definido con artefactos o módulos de software concreto, desarrollado por Twitter. Esta tecnología simplifica el proceso de creación de la interfaz de usuario combinando HTML (por sus siglas en inglés *HyperText Markup Language*, Lenguaje de Marcas de Hipertexto), JavaScript y CSS (por sus siglas en inglés *Check Cascading Style Sheets*, Hojas de Estilo en Cascada). Además se adapta a los distintos navegadores con numerosos componentes webs como: botones, etiquetas, alertas, entre muchos otros definidos en la web. Bootstrap fue programado para dar soporte a CSS 3 y HTML 5. Bootstrap está diseñado para todos los niveles: diseñador, desarrollador y principiante. Este *framework* se utiliza para hacer más fácil y rápida su implementación. Bootstrap está definido por módulos que son reutilizables e independientes en la página web y es una gran comunidad abierta y seguida por millones de personas (Bootstrap, 2013).

#### **Ventajas que proporciona:**

- Uso de HTML5 Y CSS3.
- Utiliza Web kit (plataforma para aplicaciones que funciona como base para navegadores).
- Esta todo definido por módulos que son reutilizables e independientes en la página web.
- Es una gran comunidad abierta y seguida por millones de personas.
- Se utiliza básicamente para crear interfaces más rápidas y prototipar sitios webs.
- Además existe un generador para personalizar tu propio Bootstrap aunque el que viene por defecto está muy bien diseñado (Bootstrap, 2013).

#### 1.5.4. *Framework* de desarrollo.

**Symfony:** En su versión 2.6.3 es un completo marco de trabajo diseñado para optimizar, gracias a sus características, el desarrollo de aplicaciones web. Para empezar, 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, y otros, 2010).

#### **Ventajas que proporciona:**

- Tiene su propia forma de trabajo, con variantes del MVC clásico como la capa de abstracción de base de datos, el controlador frontal y las acciones.
- Sigue la mayoría de mejores prácticas y patrones de diseño para la web.
- El completo sistema de log permite a los administradores acceder hasta el último detalle de las actividades que realiza la aplicación
- Fácil de instalar y configurar en la mayoría de plataformas de programación.
- Sencillo de usar en la mayoría de casos, pero lo suficientemente flexible como para adaptarse a los casos más complejos.

#### 1.5.5. **Servidor web.**

**Apache:** En su versión 2.2.20, es un servidor web de código abierto, altamente configurable y modular. Utiliza Perl, PHP y otros lenguajes scripts. Su función principal es analizar cualquier archivo solicitado por un navegador y mostrar resultados correctos de acuerdo con el código del archivo. Permite configurar los informes de errores, presenta visualización de códigos en numerosos niveles y la capacidad de determinar qué nivel del navegador puede aceptar el contenido. Es uno de los primeros servidores en soportar host basados en direcciones IP (por sus siglas en inglés *Internet Protocol*, Protocolo de Internet) y host virtuales. Tiene un elaborado índice de directorios, un directorio de alias, informe de errores HTTP HTML (por sus siglas en inglés *Hypertext Transfer Protocol*, Protocolo de Transferencia de Hipertexto)

configurable, gestión de recursos para procesos hijos, reescritura de las URL (por sus siglas en inglés *Uniform Resource Locator*, Localizador de Recurso Uniforme), comprobación de ortografía de las URL y manuales online (Naramore, 2005).

**Ventajas que proporciona:**

- Esta incluye formatos de configuración no estándar.
- Es multiplataforma.
- Permite elegir el servidor web Apache.
- Soporta múltiples plataformas por lo que genera mayor usabilidad, dando la opción de utilizar diferentes sistemas operativos sin ningún problema.

**1.5.6. Herramienta CASE.**

**Visual Paradigm:** Es una herramienta CASE que utiliza el lenguaje de modelado estándar UML, permite la generación de códigos e ingeniería inversa. Esta herramienta cumple con las políticas de migración a software libre en Cuba, ya que es una herramienta multiplataforma que se puede utilizar tanto en Linux como en Windows. La misma propicia un conjunto de ayudas para el desarrollo de programas informáticos, desde la planificación, pasando por el análisis y el diseño, hasta la generación del código para entornos integrados de desarrollo tales como: NetBeans, Eclipse, Oracle JDeveloper y JBuilder (Howard, 2008).

**Ventajas que proporciona:**

- Tiene una interfaz muy intuitiva y es de fácil aprendizaje para los desarrolladores.
- Permite la generación automática de diagramas a partir de descripciones de casos de usos.
- Permite hacer descripción de los casos de usos dando una gran variedad de plantillas predeterminadas permitiendo personalizarlas.

### 1.5.7. IDE de desarrollo.

**Netbeans:** En su versión 7.1, es un IDE bajo licencia GPL y de código abierto. Esta herramienta tiene la finalidad de permitirle a los desarrolladores crear diferentes sistemas y proyectos orientados sobre todo a la creación de soluciones en lenguaje Java, ya sea que se encuentren en Java SE (Edición Estándar) o Java EE (Edición Empresarial), además de soportar otros lenguajes tales como PHP y JavaScript (Jácome, y otros, 2009). Se requiere utilizar Netbeans debido a que facilita el desarrollo del sistema, al proveer un entorno de desarrollo profesional para desarrollar sistemas web.

### 1.5.8. Sistema Gestor de Base de Datos (SGBD).

**PostgreSQL:** En su versión 9.1, es uno de los motores de base de datos relacionales más potentes que existen actualmente. Permite ejecutar consultas SQL, las cuales posibilitan actualizar, insertar, eliminar y realizar reportes sobre los datos almacenados en ficheros o bases de datos. Ofrece la posibilidad de ejecutar y trabajar varios procesos al mismo tiempo sobre la misma tabla sin ser dañada, donde cada usuario obtiene una versión de lo último que ha hecho evitando la pérdida de información. Tiene su propio lenguaje PL/PgSQL, pero también se pueden usar lenguajes como C, C++, Gambas, Java PL, Java Web, Perl, Php, Python (Novella Latorre, 2012). Este gestor ofrece muchas ventajas respecto a otros sistemas de bases de datos como son:

- Mejor soporte que los proveedores comerciales: tiene una importante comunidad de profesionales.
- El código fuente está disponible para todos de manera gratuita.
- Multiplataforma: PostgreSQL está disponible en varias plataformas como son Linux y Windows.
- Notación de Modelado de Procesos de Negocio (BPMN).

### 1.5.9. Notación de Modelado de Procesos de Negocio (BPMN).

La notación BPMN (por sus siglas en inglés *Business Process Model and Notation*, Notación de Modelado de Procesos de Negocio ) en su versión 2.0, es un estándar basado en los diagramas de flujo, adaptado para suministrar una notación gráfica que describe la lógica de los pasos de un proceso de negocio a

través de flujos de trabajo. Esta notación ha sido especialmente diseñada para coordinar la secuencia de los procesos y los mensajes que fluyen entre los participantes de las diferentes actividades (Bizagi, 2012).

Por la importancia que tiene este tipo de modelado se utiliza para la modelación del proceso de gestión de la información del Plan y Presupuesto en la UCI. Es independiente de cualquier metodología de desarrollo de software, crea un puente estandarizado para disminuir la brecha entre los procesos de negocio y la implementación de éstos; permite modelar los procesos de una manera unificada y estandarizada permitiendo un entendimiento entre todas las personas de la organización.

### **1.6. Conclusiones Parciales.**

El análisis realizado a los conceptos fundamentales asociados al estudio del problema, permitió una mejor interpretación del marco teórico de la investigación. El estudio de algunas soluciones informáticas existentes en el mundo, asociadas a la gestión del Plan y Presupuesto, contribuyó a identificar funcionalidades y la forma en que se visualiza la información en las mismas. La metodología seleccionada permite proveer la eficiencia y rapidez en el proceso de desarrollo de SIGEPLAN. El estudio y selección de las herramientas seleccionadas permite obtener un sistema en correspondencia con las necesidades del cliente.

## **CAPÍTULO II: ANÁLISIS Y DISEÑO DEL SISTEMA DE GESTIÓN DE LA INFORMACIÓN DE PLAN Y PRESUPUESTO.**

### **2.1. Introducción.**

En el presente capítulo, se exponen las principales características del sistema, mediante el modelado del negocio, los requisitos funcionales y no funcionales y las historias de usuarios relacionadas a estos. En correspondencia con la etapa de planificación del sistema, se puntualiza el Plan de entregas e iteraciones. Se describe la arquitectura por la cual se rige la construcción de la solución, los patrones de diseño y los artefactos que son utilizados para cumplir con el objetivo que da solución al problema que dio origen a la presente investigación.

### **2.2. Descripción general de la propuesta de solución.**

El proceso general de propuesta de solución se centra en resolver los problemas existentes actualmente en el área de Planificación y Estadística, en cuanto a la Elaboración del Plan y el Presupuesto. Se lleva a cabo la informatización del proceso actual, de manera que la información se encuentre más segura y más fácil de manipular, esto se facilita ofreciéndoles a los usuarios los reportes que deseen, además que cada administrador de partida tendrá acceso solo a los datos con los que interactúen de forma primaria, para disminuir los errores humanos. Se garantiza que el sistema desarrollado, brinde al usuario poder exportar información contenida dentro de él, a Microsoft Excel. Todo esto permite al área de Planificación y Estadística poder llevar a cabo el proceso de una manera más organizada y menos propensa a errores. Con el objetivo de restringir el acceso a las opciones de gestión que presenta el sistema, se definen los siguientes roles:

- **Administrador general:** Posee control total por lo que puede realizar cualquier acción sobre el sistema.
- **Administrador de partidas:** Gestiona las partidas correspondientes al área que maneja.
- **Especialista:** Encargado de elaborar el Plan y Presupuesto.

- **Invitado:** Rol que se le asigna al usuario en el sistema hasta que el administrador le proporcione un rol de acuerdo a la información que maneja.

### 2.3. Modelado de procesos de negocio.

El modelado de procesos del negocio facilita entender y organizar algunos procesos que están conformados de múltiples actividades, personas y departamentos en una organización (Benghazi, y otros, 2010). Aunque la metodología seleccionada, no propone este diagrama, su uso trae como beneficios conocer cómo se realiza el proceso de principio a fin, revelar las fallas y problemas, así como generar soluciones a los problemas detectados. Se realiza mediante la notación BPMN, a través de la cual se puede realizar diferentes niveles de procesos.

La Figura 1 muestra el macroproceso Elaboración del Plan y Presupuesto, el cual tiene como subprocesos: Llenar el modelo de las partidas correspondientes a cada área, Emitir el modelo del Plan por área y Conformar los datos del modelo del Plan por partidas asociadas a cada área.

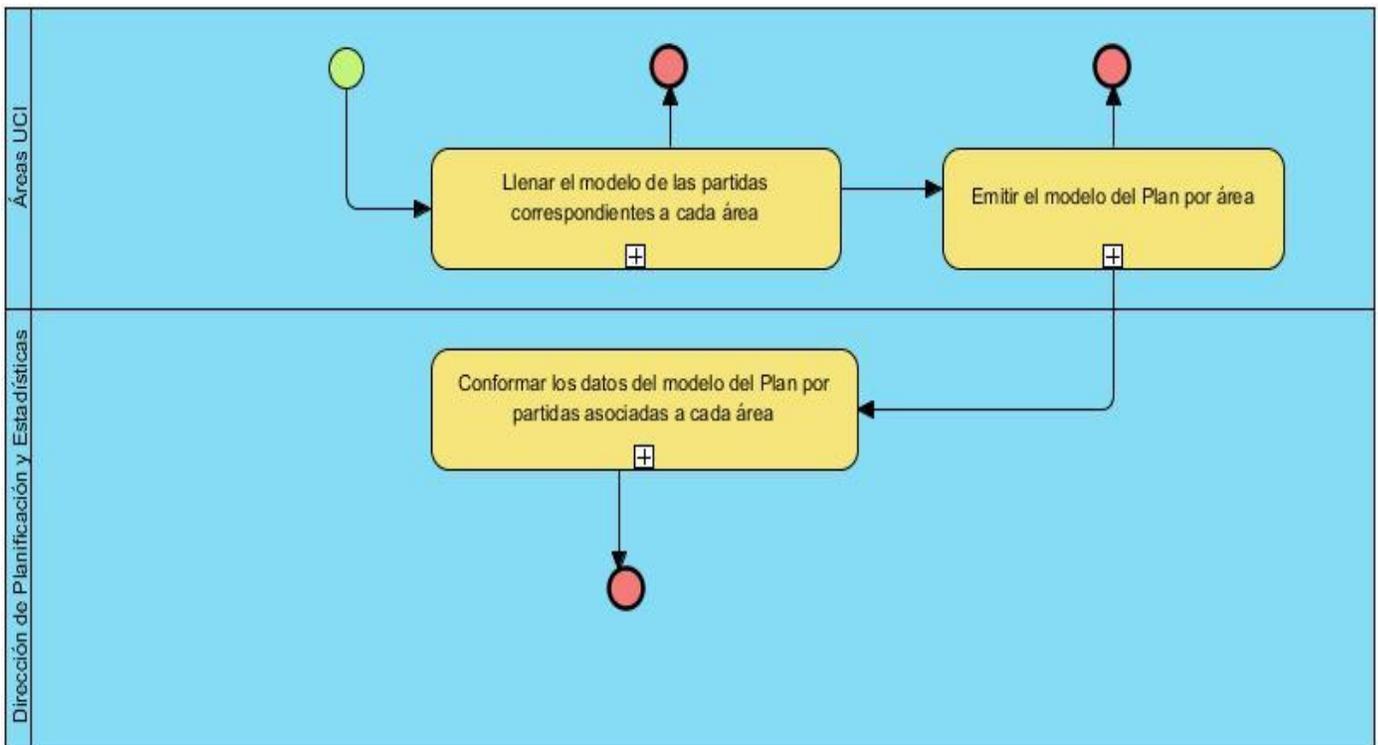


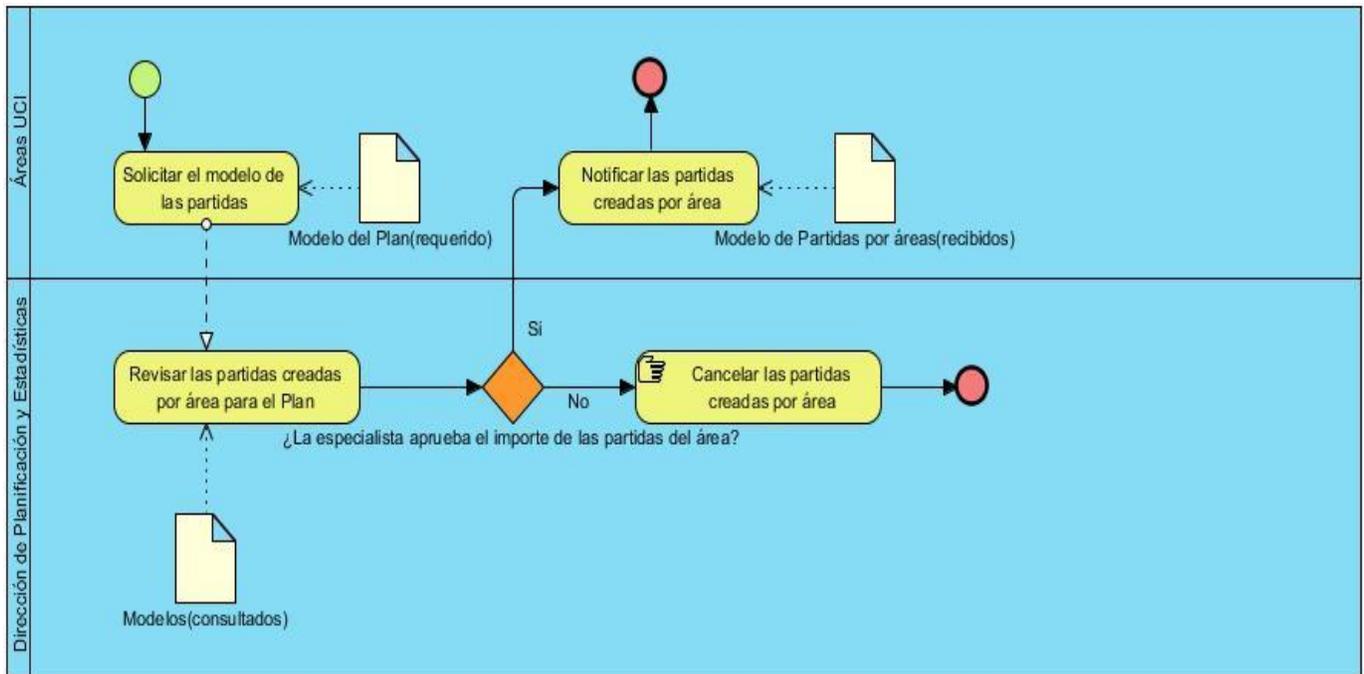
Figura 1: Proceso de negocio. Elaboración del Plan y Presupuesto.

## **Proceso de negocio de la Elaboración del Plan y el Presupuesto.**

El proceso del Plan y Presupuesto se ejecuta en todas las entidades desde su creación como empresa, entidad u organismo. Este se ejecuta de forma tal, que permite conocer los Planes que se trazan cada área con el fin de llegar a conocer el Presupuesto que pueden ejecutar de forma anual. Este se lleva a cabo a través de modelos que permiten tener un mejor control de los datos que se quieren certificar para su uso. Estos modelos tienen designado un grupo de partidas de gastos que nos son más que los conceptos a los cuales se les va a asignar un dinero para después poder usarse, estas partidas no todas son iguales para todas las áreas, cada una de ellas, dependen de las ejecuciones y necesidades que tenga cada área de la UCI. A continuación se presenta cada uno de los subprocesos mostrados con las actividades que lo conforman por lo que facilita un mejor análisis y comprensión de los mismos.

### **2.3.1. Subproceso: Llenar el modelo de las partidas correspondientes a cada área.**

Los administradores de partidas son los encargados de llenar los modelos correspondientes a sus respectivas áreas en aras de elaborar el Presupuesto anual. Luego la Dirección de Planificación y Estadística se encarga de revisar las partidas que fueron creadas y el llenado de las mismas por parte de las áreas, en caso de estar de acuerdo, esta Dirección notifica que las partidas creadas están bien y que los datos que las conforman están en perfecto estado. Si los datos insertados por las áreas están mal, entonces se procede por parte de la Dirección a cancelar los datos que fueron insertados. La Figura 2 muestra el desempeño de cada una de las actividades:



**Figura 2: Proceso de negocio: Llenar los modelos de las partidas correspondientes a cada área.**

### 2.3.2. Subproceso: Emitir el modelo del Plan por área.

Este proceso comienza cuando los jefes de partidas, que son los que representan las áreas de la UCI, llenan los modelos de las partidas que tienen asociadas para ser usadas de forma anual, luego de aprobarse el Presupuesto que se solicite mediante las partidas. Una vez ejecutada esta actividad, la Dirección de Planificación y Estadística se encarga de enviar los modelos elaborados por las áreas a estas luego de ser revisados los datos llenados por ellos. Si este proceso luego de ser revisado está bien, se le notifica a las áreas la aceptación de los datos insertados, los jefes de áreas reciben entonces la notificación. En caso de que estos datos luego de revisados están mal, se procede a cancelarlos por parte de la Dirección de Planificación y Estadística y se le notifica a las áreas para que estas vuelvan a modificar los datos con las sugerencias que se le hace por parte de la Dirección y luego entonces de llenados por las áreas UCI, se procede a revisar de nuevo por la Dirección y se ejecuta el proceso de notificación al área que los nuevos datos están bien y de esta forma cierra el ciclo.

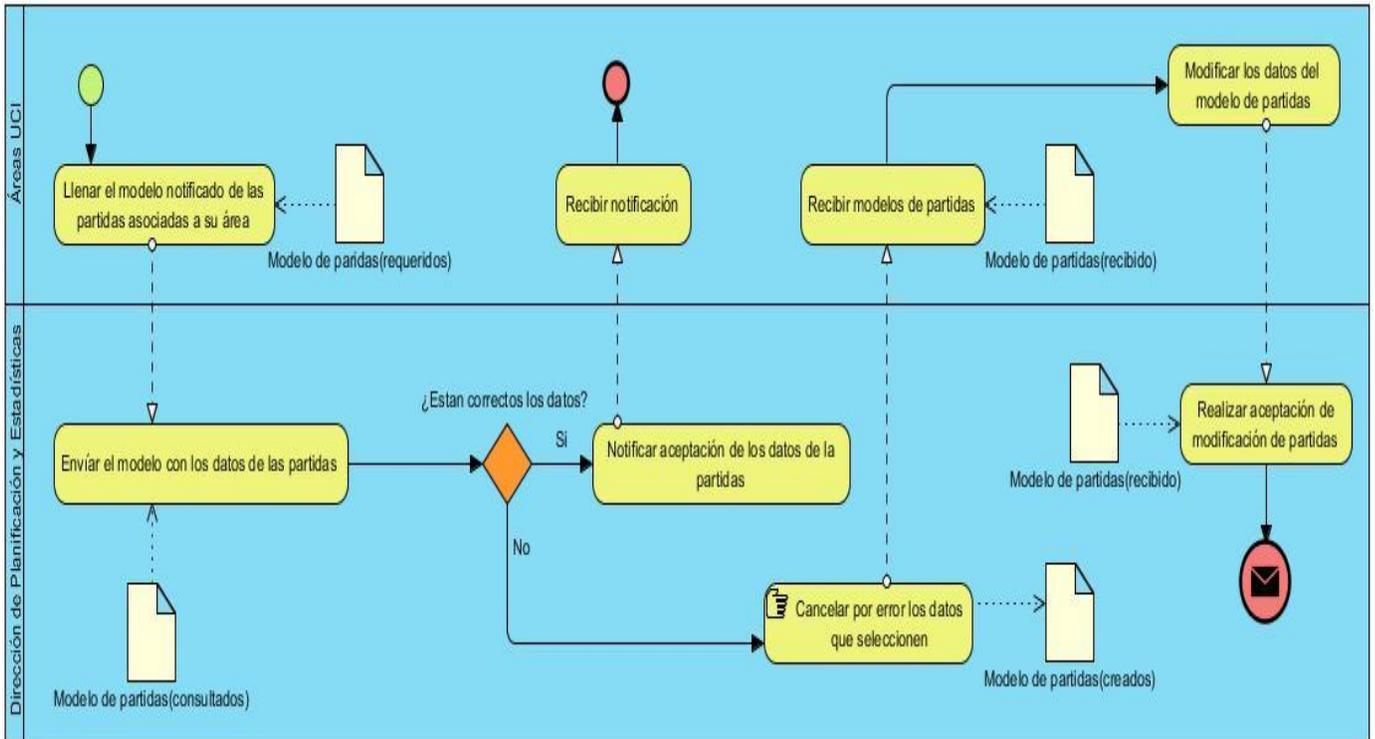


Figura 3: Proceso de negocio. Emitir el modelo del Plan por área.

### 2.3.3. Subproceso: Conformar los datos del modelo del Plan por partidas asociadas a cada área.

Este proceso comienza cuando el área de la Dirección de Planificación y Estadística se encarga de recibir los datos que fueron llenados por las áreas luego de haberse revisado antes y se procede a recoger estos datos insertados, logrando así aceptar los datos insertados por las áreas en el sistema, luego entonces esta Dirección se encarga de notificar que las partidas que se llenaron por las áreas están bien y que estas conforman el Plan del área y que serían estas las únicas partidas que pueden ser usadas por el área.

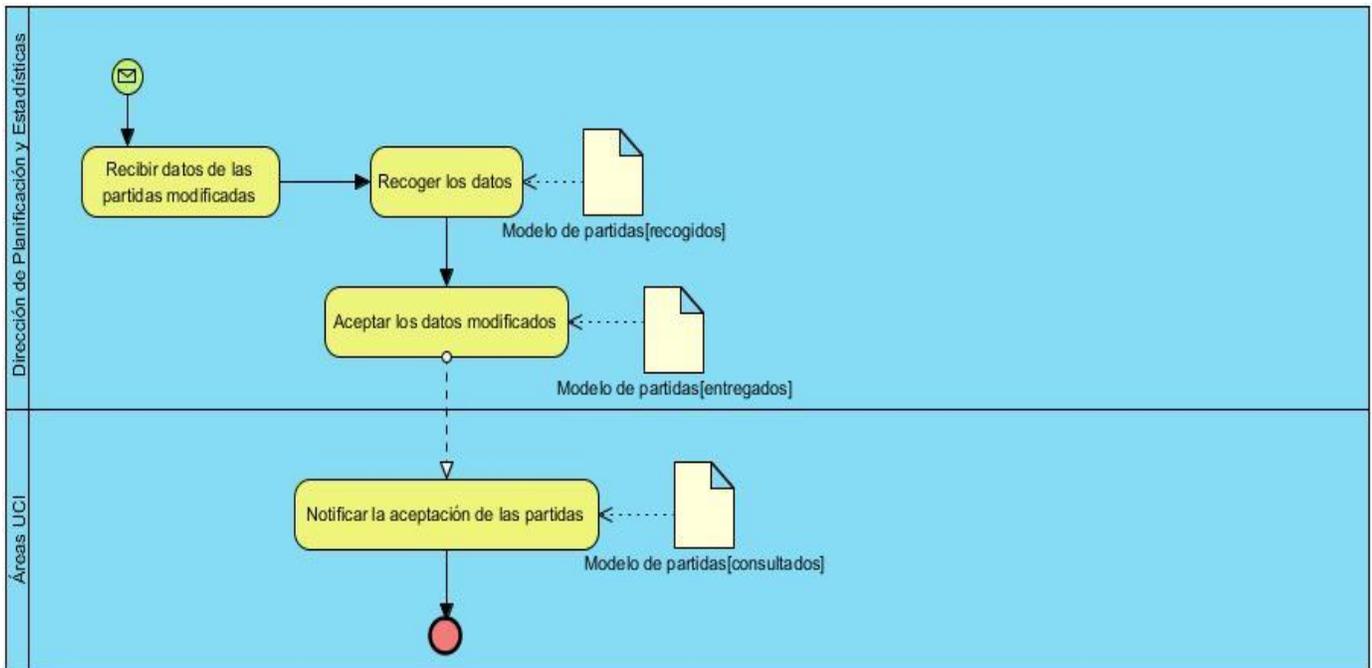


Figura 4: Proceso de negocio. Conformar los datos del modelo del Plan por partidas asociadas a cada área.

#### 2.3.4. Roles que intervienen en el proceso de negocio.

**Jefes de Partidas:** Son los encargados de entrar al sistema los datos que se refieran a las partidas correspondientes a sus áreas, con un grupo de justificantes que demuestren los datos entrados.

**Especialistas de la Dirección de Planificación y Estadística:** Los especialistas son los encargados de revisar y aprobar cada uno de los datos entrados por cada uno de los jefes de partidas, además de ser los encargados de notificar el Plan para cada año de las áreas.

#### 2.4. Requisitos del sistema.

Los requisitos son la descripción de los servicios proporcionados por el sistema y sus restricciones operativas. El flujo de trabajo de requisitos es uno de los más importantes para el desarrollo de un software, ya que su propósito fundamental es guiar el desarrollo hacia el sistema correcto (Torres Almira, 2012). Dentro de sus varias clasificaciones se encuentra funcionales o no funcionales. Para la adquisición de requisitos se emplean algunas técnicas como entrevistas, cuestionarios y tormenta de ideas.

La técnica seleccionada para el levantamiento de los requisitos del sistema a desarrollar fue la entrevista, la cual se le realizó a: la Directora de Planificación y Estadística de la UCI, así como a la mayoría de las especialistas que laboran en esa misma área.

#### 2.4.1. Requisitos funcionales (RF).

Describen las interacciones entre el sistema y su ambiente, en forma independiente a su implementación. El ambiente incluye al usuario y cualquier otro sistema externo con el cual interactúe el sistema (Sommerville, 2002). La técnica utilizada para la validación de los RF fue los Prototipos de Interfaz los cuales fueron mostrados a los clientes con el fin de que aprobaran la propuesta. En la tabla 1 se muestran los RF:

| Listado de RF                       |  |
|-------------------------------------|--|
| RF-1. Adicionar bloque.             | RF-15. Eliminar datos de partidas.       |
| RF-2. Editar bloque.                | RF-16. Mostrar datos de partidas         |
| RF-3. Mostrar bloque.               | RF-17. Buscar partida por área.          |
| RF-4. Eliminar bloque.              | RF-18. Adicionar planificación.          |
| RF-5. Adicionar grupo.              | RF-19. Editar planificación.             |
| RF-6. Editar grupo.                 | RF-20. Mostrar planificación.            |
| RF-7. Mostrar grupo.                | RF-21. Eliminar planificación.           |
| RF-8. Eliminar grupo.               | RF-22. Mostrar Presupuesto.              |
| RF-9. Adicionar asignación.         | RF-23. Editar Presupuesto.               |
| RF-10. Mostrar asignación.          | RF-24. Exportar reportes en formato xls. |
| RF-11. Editar asignación.           | RF-25. Autenticar usuario.               |
| RF-12. Eliminar asignación.         | RF-26. Editar roles de usuarios.         |
| RF-13. Adicionar datos de partidas. | RF-27. Mostrar roles de usuarios         |
| RF-14. Editar datos de partidas.    | RF-28. Eliminar usuario del sistema.     |

Tabla 1: Listado de requisitos funcionales.

#### 2.4.2. Requisitos no funcionales (RNF).

Describen atributos sólo del sistema o del ambiente del sistema que no están relacionados directamente con los RF. Los RNF incluyen restricciones cuantitativas, como el tiempo de respuesta o precisión, lenguajes de programación y/o sistemas operativos, etcétera (Sommerville, 2002). Se definieron como RNF los siguientes:

- **Usabilidad:**

**RNF-1.** El sistema debe mostrar un menú en la parte lateral izquierda con las funcionalidades del sistema.

**RNF-2.** La iconografía usada es estándar para cada acción del sistema.

**RNF-3.** Si se cometen errores el sistema muestra un mensaje de error de color rojo.

- **Software:**

**RNF-4.** Las computadoras de los clientes sólo requieren de un navegador Mozilla Firefox o Google Chrome en su versión 8.0 y 10.0 respectivamente o superior.

**RNF-5.** El servidor de aplicaciones debe tener el sistema operativo CentOS 7 o superior.

**RNF-6.** El Gestor de base de datos a utilizar es PostgreSQL en su versión 9.1.

**RNF-7.** El servidor de aplicaciones debe tener como servidor web Nginx en su versión 1.0.15 o superior.

- **Seguridad:**

**RNF-8.** Se garantiza la integridad y confidencialidad de la información mediante mecanismos de control de acceso no autorizados utilizando: usuario, contraseña y definiendo niveles de acceso para cada usuario, de manera que cada usuario pueda tener disponible solamente las opciones relacionadas con su actividad y tenga datos de acceso propios.

**RNF-9.** Se podrá acceder a las páginas de administración del sitio web y las páginas de usuarios a través del protocolo https.

**RNF-10.** El sistema garantiza que al eliminar información pueda existir una opción de advertencia antes realizar la acción.

**RNF-11.** Los mensajes de error mostrados a los usuarios deben ser genéricos sin dar detalles de la información, para no comprometer la seguridad e integridad de los datos.

- **Hardware:**

**RNF-13.** Los requerimientos mínimos para las computadoras de los clientes son el uso de un microprocesador Pentium 4, con 512 MB de RAM, 80 GB de disco duro y con acceso a la red UCI.

**RNF-14.** El servidor de base de datos debe tener como tecnología mínima un microprocesador Core 2duo, con 2 GB de RAM y 80 GB de disco duro.

**RNF-15.** El servidor de aplicaciones debe ser Core 2duo, con 2 GB de RAM y 80 GB de disco duro.

- **Soporte:**

**RNF-16.** El sistema debe ser instalado y probado donde lo necesite el cliente.

## **2.5. Planificación.**

### **2.5.1. Historia de usuario (HU).**

Las HU son la técnica utilizada en XP para especificar los requisitos del software. Estas describen brevemente las características que el sistema debe poseer. Cada HU debe ser lo suficientemente comprensible y delimitada para que los programadores puedan implementarla. Estas se usan para estimar el tiempo y el Plan de entregas los cuales dirigen la creación de las pruebas de aceptación. Se caracterizan por ser independientes una de otras, negociables, valoradas por los clientes o usuarios, estimables y verificables (Letelier, y otros, 2006).

Las estimaciones de esfuerzo asociado a la implementación de las HU la establecen los programadores, utilizando como medida el punto. Un punto, equivale a una semana ideal de programación. Las historias generalmente valen de 1 a 3 puntos (Letelier, y otros, 2006). A continuación se presentan las principales HU, las restantes se presentan en el Anexo 3.

| <b>HISTORIA DE USUARIO</b>                                 |                                    |
|--|------------------------------------|
| <b>Número:</b> HU#12                                       | <b>Nombre:</b> Autenticar usuario. |
| <b>Modificación de historia de usuario Número:</b> Ninguna |                                    |
| <b>Usuario:</b> Jorge Pozo Hechavarria                     | <b>Iteración asignada:</b> 2       |
| <b>Prioridad en Negocio:</b> Alta                          | <b>Puntos Estimados:</b> 1 semana  |
| <b>Riesgo en Desarrollo:</b> Medio                         | <b>Puntos Reales:</b> 5 días       |

**Descripción:** La HU inicia cuando el usuario se dirige a la opción “Inicio de sesión”. Para que el usuario pueda autenticarse en el sistema debe entrar los siguientes datos:

- Usuario (obligatorio, campo de texto)
- Contraseña (obligatorio, campo de texto)

Luego dar clic en la opción: “Iniciar Sesión”, finalizando así la HU.

**Observaciones:**

- Si el usuario introduce la información dejando campos obligatorios vacíos, el sistema emite un mensaje indicándole que los campos obligatorios deben de llenarse.
- El usuario debe autenticarse con usuario y contraseña de dominio de (LDAP).
- Si el usuario introduce la información de forma incorrecta, el sistema emite un mensaje notificando el error.

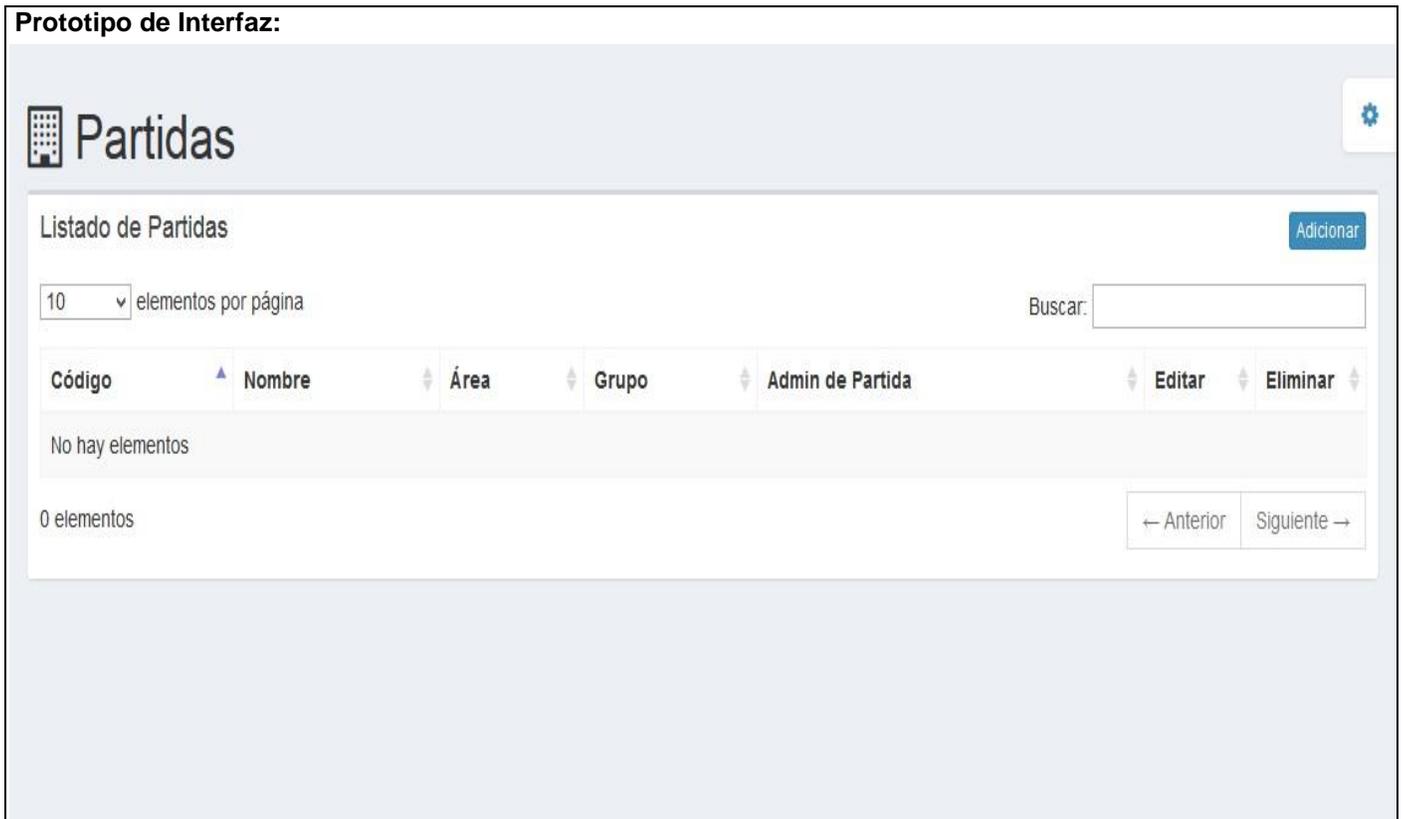
**Prototipo de Interfaz:**



**Tabla 2: HU Autenticar usuario.**

| HISTORIA DE USUARIO  |  |
|--|--|
| <b>Número:</b> HU#4  | <b>Nombre:</b> Gestionar datos de partidas |
| <b>Modificación de historia de usuario Número:</b> Ninguna   |  |
| <b>Usuario:</b> Rolando Ortiz Sánchez  | <b>Iteración asignada:</b> 1               |
| <b>Prioridad en Negocio:</b> Alta  | <b>Puntos Estimados:</b> 1 semana          |
| <b>Riesgo en Desarrollo:</b> Alto  | <b>Puntos Reales:</b> 1 semana             |
| <b>Descripción:</b> Permite adicionar, mostrar, editar y eliminar los datos de las partidas para la Elaboración del Plan y Presupuesto.  |  |
| <b>Observaciones:</b> Los campos a rellenar estarán validados de acuerdo a los tipos de datos correspondientes en cada uno, si es texto no podrá introducir otra cosa que no sea texto. Cuando se ejecuta la funcionalidad eliminar, el sistema muestra una notificación para preguntar al usuario si está seguro de eliminar los datos. |  |

**Prototipo de Interfaz:**



**Tabla 3: HU Gestionar datos de partidas.**

**2.5.2. Plan de entregas.**

El Plan de entregas es el artefacto siguiente que propone la metodología antes mencionada. En esta fase el cliente establece la prioridad de cada HU, el programador realiza una estimación del esfuerzo necesario de cada una de ellas. Se toman acuerdos sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente. Se propone el siguiente Plan de entregas para la solución propuesta:

| Entregables      | Iteración | Fin de la Iteración |
|------------------|-----------|---------------------|
| Adicionar bloque | 1         | Febrero/2015        |
| Editar bloque    | 1         | Febrero/2015        |
| Mostrar bloque   | 1         | Febrero/2015        |
| Eliminar bloque  | 1         | Febrero/2015        |
| Adicionar grupo  | 1         | Febrero/2015        |
| Editar grupo     | 1         | Febrero/2015        |
| Mostrar grupo    | 1         | Febrero/2015        |

|                                   |   |              |
|-----------------------------------|---|--------------|
| Eliminar grupo                    | 1 | Febrero/2015 |
| Adicionar asignación              | 1 | Febrero/2015 |
| Editar asignación                 | 1 | Febrero/2015 |
| Mostrar asignación                | 1 | Febrero/2015 |
| Eliminar asignación               | 1 | Febrero/2015 |
| Adicionar datos de partidas       | 1 | Febrero/2015 |
| Editar datos de partidas          | 1 | Febrero/2015 |
| Mostrar datos de partidas         | 1 | Febrero/2015 |
| Eliminar datos de partidas        | 1 | Febrero/2015 |
| Buscar partida por área           | 2 | Marzo/2015   |
| Adicionar planificación           | 2 | Marzo/2015   |
| Editar planificación              | 2 | Marzo/2015   |
| Mostrar planificación             | 2 | Marzo/2015   |
| Eliminar planificación            | 2 | Marzo/2015   |
| Mostrar Presupuesto               | 2 | Marzo/2015   |
| Editar Presupuesto                | 2 | Abril/2015   |
| Exportar reportes en formato xls. | 2 | Abril/2015   |
| Autenticar usuario                | 2 | Abril/2015   |
| Editar roles de usuario           | 2 | Abril/2015   |
| Mostrar roles de usuario          | 2 | Mayo/2015    |
| Eliminar usuario del sistema      | 2 | Mayo/2015    |

Tabla 4: Plan de entrega.

### 2.5.3. Plan de iteraciones.

El ciclo de desarrollo de software guiado por XP se caracteriza por ser iterativo e incremental, por lo que se realizan varias iteraciones sobre el sistema antes de su fase de pruebas. Para elaborar el Plan de iteraciones se deben tomar en cuenta las HU no abordadas, velocidad del proyecto, pruebas de aceptación no superadas en la iteración anterior y tareas no terminadas (Letelier, y otros, 2006).

El Sistema de Gestión de la Información referente al Plan y Presupuesto se consultó con el cliente el cual establece realizarlo en dos iteraciones:

**Primera iteración:** Se desarrollan las principales HU para el funcionamiento básico del sistema; es decir, todo lo relacionado con partidas de gastos, Grupo, Asignación y Bloque como se muestra en la Tabla 3.

**Segunda iteración:** Se desarrollan las HU restantes encargadas de la seguridad, Presupuesto, los reportes y las que tienen prioridad del negocio media.

| Iteración | Historias de usuario | Semanas Estimadas |
|-----------|----------------------|-------------------|
|-----------|----------------------|-------------------|

|                              |                                   |    |
|------------------------------|-----------------------------------|----|
| 1                            | Adicionar bloque                  | 4  |
|                              | Editar bloque                     |    |
|                              | Mostar bloque                     |    |
|                              | Eliminar bloque                   |    |
|                              | Adicionar grupo                   |    |
|                              | Editar grupo                      |    |
|                              | Mostrar grupo                     |    |
|                              | Eliminar grupo                    |    |
|                              | Adicionar asignación              |    |
|                              | Editar asignación                 |    |
|                              | Mostrar asignación                |    |
|                              | Eliminar asignación               |    |
|                              | Adicionar datos de partidas       |    |
|                              | Editar datos de partidas          |    |
|                              | Eliminar datos de partidas        |    |
| Mostrar datos de partidas    |                                   |    |
| 2                            | Buscar partida por área           | 11 |
|                              | Adicionar planificación           |    |
|                              | Editar planificación              |    |
|                              | Mostrar planificación             |    |
|                              | Eliminar planificación            |    |
|                              | Mostrar Presupuesto               |    |
|                              | Editar Presupuesto                |    |
|                              | Exportar reportes en formato xls. |    |
|                              | Autenticar usuario                |    |
|                              | Editar roles de usuario           |    |
|                              | Mostrar roles de usuario          |    |
| Eliminar usuario del sistema |                                   |    |

**Tabla 5: Plan de iteraciones.**

## 2.6. Diseño.

El papel del diseño en el ciclo de vida del software es adquirir conocimiento de su funcionamiento, constituye el punto de partida para las actividades de implementación, dando soporte a los requisitos funcionales y restricciones relacionadas con los lenguajes de programación, componentes reutilizables y sistemas operativos, que debe poseer la aplicación (Torres Almira, 2012). En esta fase la metodología XP sugiere un diseño fácilmente entendible que a la larga costará menos tiempo y esfuerzo desarrollar (Oswaldo, y otros).

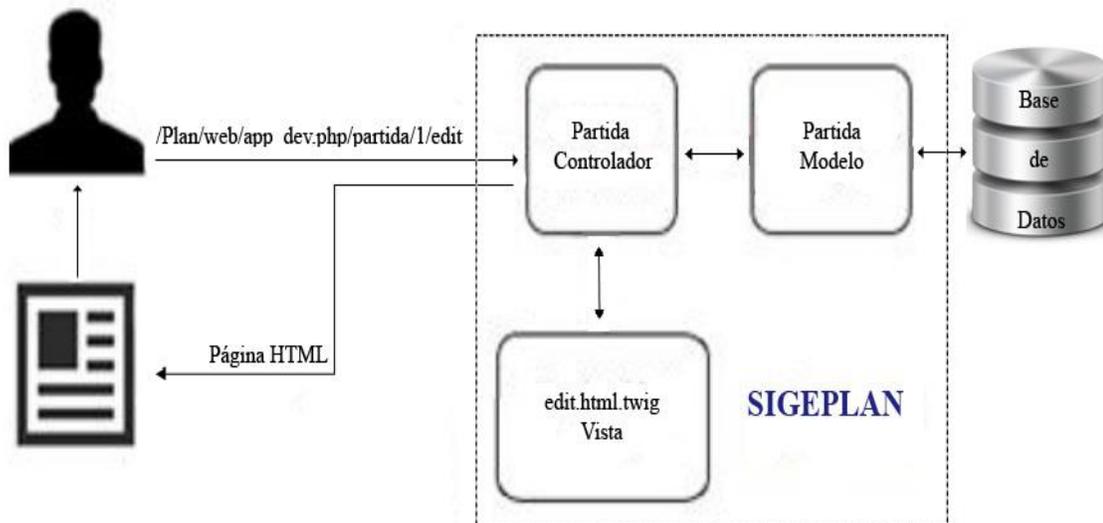
### 2.6.1. Descripción de la arquitectura.

Uno de los pasos fundamentales en el proceso de desarrollo de software es ciertamente la definición de la arquitectura del *software*, la cual constituye el diseño de más alto nivel de la organización de un sistema,

programa o aplicación. Es de gran importancia ya que la manera en que se estructura un sistema tiene un impacto directo sobre la capacidad de este para satisfacer lo que se conoce como los atributos de calidad del sistema. Ejemplos de atributos de calidad son el desempeño, que tiene que ver con el tiempo de respuesta del sistema a las peticiones que se le hacen; la usabilidad, que tiene que ver con qué tan sencillo les resulta a los usuarios realizar operaciones con el sistema (Cervantes, 2010). La arquitectura entre sus objetivos define los módulos principales del sistema, la responsabilidad que tendrá cada uno de los módulos, la interacción entre los mismos y controlar el flujo de los datos en el sistema.

La arquitectura propuesta para el desarrollo de SIGEPLAN se basa en la arquitectura del *framework* Symfony, seleccionado en el capítulo anterior. Esta se sustenta en los principios fundamentales de la arquitectura Modelo-Vista-Controlador, la cual plantea dividir y organizar el código de acuerdo a las convenciones establecidas por el *framework*. El código de la presentación se guarda en la vista, el código de manipulación de datos se guarda en el modelo y la lógica de procesamiento de las peticiones constituye el controlador (Ecured, 2015).

En la Figura 5 se evidencia la utilización del patrón mencionado anteriormente a través del esquema de funcionamiento del requisito Editar datos de partidas.



**Figura 5: Esquema de funcionamiento del requisito Editar datos de partidas.**

El sistema de enrutamiento de Synfony determina que el controlador que debe ejecutar el sistema es “Partida”, entregándole junto con la petición una variable que contiene el identificador de partida que el usuario desea modificar. Posteriormente el controlador le solicita al modelo los datos registrados en la base de datos y se los entrega a la vista para que cree una página a partir del motor de plantillas Twig. Por último, el controlador retorna dicha página, mostrándole al usuario la petición deseada.

### 2.6.2. Patrones de diseño.

Los patrones del diseño son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces. El diseño de la propuesta de solución se realizó aplicando los patrones GRASP (por sus siglas en inglés, *General Responsibility Assignment Software Patterns*), los cuales describen los principios fundamentales de la asignación de responsabilidades a objetos y GOF (por sus siglas en inglés, *Gang of Four*), los cuales definen estructuras de clases que tratan una situación en particular.

#### Los patrones GRASP utilizados fueron:

**Experto (*Expert*):** Se utiliza para la asignación de responsabilidades a las clases de forma tal que las mismas contengan la información necesaria para poder ejecutar una acción específica, o sea, el controlador frontal le asigna responsabilidades a las clases controladoras para realizar acciones de acuerdo a las peticiones que recibe. Este patrón se evidencia en las clases Partida.php, Grupo.php y Presupuesto.php.

**Creador (*Creator*):** Se utiliza para la asignación de responsabilidades a las clases relacionadas con la creación de objetos, de forma tal que una instancia de un objeto sólo pueda ser creada por el objeto que contiene la información necesaria para ello. Este patrón se evidencia en la clase PartidaController.php.

**Alta cohesión (*High Cohesion*):** Se utiliza para realizar un diseño que evite contener clases con un alto grado de abstracción, que asuman responsabilidades que podían haber delegado a otros objetos o que tengan responsabilidades muy complejas. Se diseñaron las clases de forma tal que contengan las mínimas responsabilidades necesarias y colaboren con otras para llevar a cabo una tarea. Este patrón se evidencia en las clases Asignación.php y Meses.php.

**Bajo Acoplamiento (*Low Coupling*):** El acoplamiento mide la fuerza con que una clase está conectada a otra, de esta forma una clase con bajo acoplamiento debe tener un número mínimo de dependencia con otras clases. Este patrón se evidencia en las clases Presupuesto.php y Planificación.php.

**Controlador:** Asignar la responsabilidad de controlar el flujo de eventos del sistema a clases específicas. Esto facilita la centralización de actividades (validaciones, seguridad). El controlador no realiza estas actividades, las delega en otras clases con las que mantiene un modelo de alta cohesión. Un error muy común es asignarle demasiada responsabilidad y alto nivel de acoplamiento con el resto de los componentes del sistema. Este patrón se evidencia en la clase PresupuestoController.php.

**Los patrones GOF utilizados fueron:**

**Observador (*Observer*):** Es el encargado de definir una dependencia de uno a muchos entre objetos, de forma que cuando un objeto cambie de estado se notifique y se actualicen automáticamente todos los objetos que dependen de él. Este patrón se evidencia en las clases Bloque.php y Grupo.php

**Decorador (*Decorator*):** Symfony presenta el denominado archivo base.html.twig o también conocido como plantilla global, en la que convergen todos los elementos comunes a cada una de las páginas del sistema en construcción. Este fichero se complementa con las plantillas, decorándolas y obteniéndose la interfaz final que será mostrada al usuario.

**Instancia única (*Singleton*):** Garantiza que una clase sólo tenga una instancia y proporciona un punto de acceso global a ésta instancia. Esto se evidencia en el uso del servicio *sfRouting* que es el sistema de enrutamiento de Symfony.

**Controlador frontal (*Front Controller*):** Es el único punto de entrada a la aplicación. Carga la configuración y determina la acción a ejecutarse. Se encarga de despachar las peticiones, lo que implica algo más que detectar la acción que se ejecuta. Esto se evidencia en la clase App.php.

### 2.6.3. Diseño de la base de datos.

El diseño de la base de datos representa el tratamiento de la información con carácter persistente dentro del sistema. Para este trabajo fue construido el Modelo Físico de Datos, cual proporciona una flexibilidad óptima para el soporte de la informatización.

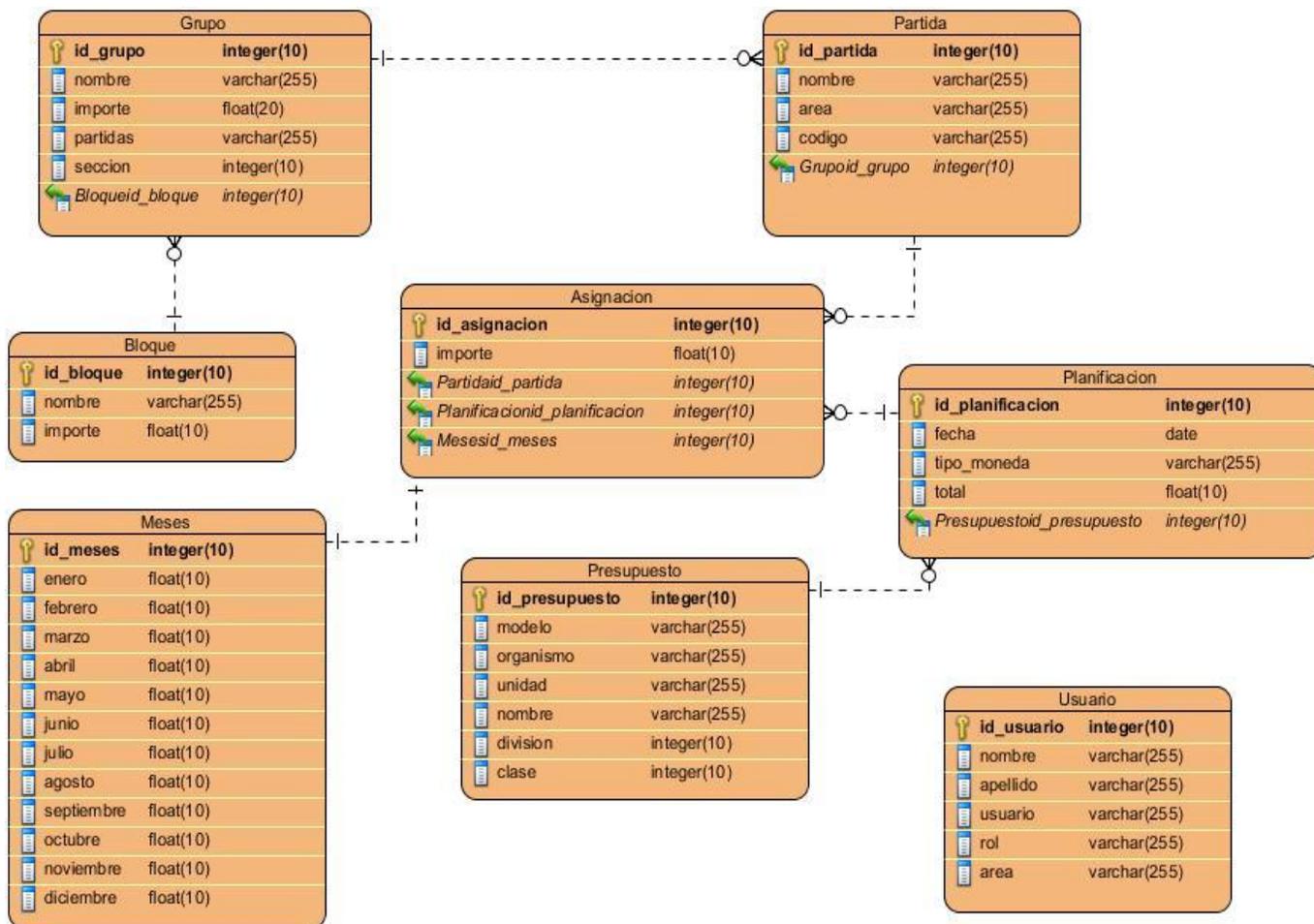


Figura 6: Modelo de Datos.

### 2.6.4. Tarjetas CRC.

La técnica CRC (por sus siglas en inglés *Class, Responsibility and Collaboration*, Clase, Responsabilidad y Colaboración) propone una forma de trabajo, preferentemente grupal, para encontrar los objetos del

dominio del dominio de la aplicación, sus responsabilidades y como colaboran con otros para realizar tareas. La técnica CRC aumenta la probabilidad de que un grupo de trabajo haga un buen análisis del sistema. CRC ayuda a identificar las clases que participan en el diseño del sistema, obtener las responsabilidades que debe cumplir cada clase y establecer como colabora una clase con otras clases para cumplir con sus responsabilidades (Diego, 2002). A continuación se presentan algunas tarjetas CRC pertenecientes al sistema, las restantes se encuentran en el Anexo 4.

| <b>Nombre de la clase: Partida</b> |   |
|------------------------------------|---|
| <b>Responsabilidades</b>           | <ol style="list-style-type: none"> <li>1. getID()</li> <li>2. setNombre(\$nombre)</li> <li>3. getNombre()</li> <li>4. setArea(\$area)</li> <li>5. getArea()</li> <li>6. setCodigo(\$codigo)</li> <li>7. getCodigo()</li> <li>8. toString()</li> <li>9. getAsignaciones()</li> <li>10. getGrupo()</li> </ol> |
| <b>Colaboradores</b>               | <ol style="list-style-type: none"> <li>1. Grupo</li> <li>2. Asignación</li> </ol>   |

**Tabla 6: Tarjeta CRC correspondiente a la clase Partida.**

| <b>Nombre de la clase: Grupo</b> |   |
|----------------------------------|---|
| <b>Responsabilidades</b>         | <ol style="list-style-type: none"> <li>1. getID()</li> <li>2. setNombre(\$nombre)</li> <li>3. getNombre()</li> <li>4. setImporte(\$importe)</li> <li>5. getImporte()</li> <li>6. getPartida()</li> <li>7. setSeccion(\$seccion)</li> <li>8. getSeccion()</li> </ol> |
| <b>Colaboradores</b>             | <ol style="list-style-type: none"> <li>1. Partida</li> <li>2. Bloque</li> </ol>   |

**Tabla 7: Tarjeta CRC correspondiente a la clase Grupo.**

## **2.7. Conclusiones Parciales.**

El modelado de los procesos del negocio y la explicación detallada de estos, permitieron identificar con mayor facilidad los requisitos funcionales que el sistema debe satisfacer, propiciando así la comprensión de su funcionamiento. La confección de diversos artefactos pertenecientes a la metodología utilizada como: las Historia de Usuario y las Tarjetas CRC permitieron agilizar y organizar el proceso de desarrollo

del sistema. El uso de patrones y el empleo del *framework* Symfony permitió obtener un sistema mantenible y escalable.

## **CAPÍTULO III: IMPLEMENTACIÓN Y PRUEBAS DEL SISTEMA DE GESTIÓN DE LA INFORMACIÓN DE PLAN Y PRESUPUESTO.**

### **3.1. Introducción.**

El capítulo presenta los resultados de la implementación y validación; es decir, se describe cómo realizar una correcta implementación a partir del análisis y diseño, además se discuten los resultados alcanzados al aplicar las pruebas de caja blanca y de aceptación. Para ello se especifican los estándares de codificación que debe cumplir el equipo de desarrollo, se crea el diagrama de despliegue representando los recursos físicos necesarios para el despliegue de la aplicación, y se presentan las interfaces de pruebas creadas para comprobar el funcionamiento de la solución. Además se realiza la validación de las variables de la investigación.

### **3.2. Estándares de codificación.**

Los estilos o estándares de codificación son pautas de programación que están enfocadas en la estructura y apariencia física del programa, facilitando la comprensión del código por terceras personas. Su uso permite reflejar un estilo armonioso, como si todo el código hubiese sido escrito por un solo programador. Si se aplica de manera continua un estándar de codificación y posteriormente se realizan revisiones del código, aumentan las posibilidades de que el proyecto de software se convierta en un sistema fácil de comprender y de mantener (Hristov, 2007). A continuación se define el estilo de codificación utilizado en la aplicación:

#### **Etiquetas de apertura y cierre de PHP**

- Siempre se debe utilizar las etiquetas de apertura `<?php`.

#### **Operadores**

- Los operadores binarios, que se utilizan entre dos valores, deben separarse de estos valores, a ambos lados del operador, por un espacio.

### **Uso del punto y coma en PHP**

- Cada línea de código independiente debe finalizar con un punto y coma (;).

### **Funciones**

- Los nombres de las funciones deben estar escritos en minúsculas y las palabras separadas por guion bajo. Además, se debe incluir siempre como prefijo el nombre del módulo, tema, para evitar así duplicidad de funciones. En su declaración, después del nombre de la función, el paréntesis de inicio de los argumentos debe ir sin espacio. Cada argumento debe ir separado por un espacio, después de la coma del argumento anterior. En la llamada a la función se aplican las mismas reglas anteriores con respecto a los parámetros. Como excepción, es posible usar más de un espacio antes de una asignación (=) para mejorar la presentación, cuando se estén realizando varias asignaciones en bloque.

### **Arrays**

- Los valores dentro de un array (o matriz) se deben separar por un espacio (después de la coma que los separa). El operador => debe separarse por un espacio a ambos lados. Cuando la línea de declaración del array supera los 80 caracteres, cada elemento se debe escribir en una única línea, indentándolo una vez (2 espacios). En este último caso, la coma de separación del último elemento también se escribirá, aunque no existan más elementos. De esta forma se evitan errores al añadir nuevos elementos al vector.

### **Comentar el código**

- Se pueden utilizar las etiquetas /\* \*/ para comentarios en varias líneas y // para comentarios de una única línea. Se deben escribir frases completas, comenzándolas con mayúscula y terminándolas con un punto. En caso de que en el comentario se haga referencia a una constante, ésta debe escribirse en mayúsculas (por ejemplo TRUE o FALSE).

En la Figura 7 se presenta un fragmento de código para ejemplificar algunos de los estándares:

```
45 public function createAction(Request $request)
46 {
47     $entity = new Partida();
48     $form = $this->createForm($entity);
49     $form->handleRequest($request);
50
51     if ($form->isValid()) {
52         $em = $this->getDoctrine()->getManager();
53         $em->persist($entity);
54         $em->flush();
55
56         return $this->redirect($this->generateUrl('partida'));
57     }
58
59     return array(
60         'entity' => $entity,
61         'form'   => $form->createView(),
62     );
63 }
```

Figura 7: Fragmento de Código.

### 3.3. Implementación.

La implementación como fase del proceso de desarrollo de software, se nutre de los resultados obtenidos en el diseño. El modelo de implementación contiene el diagrama de despliegue. Este describe la dependencia entre nodos físicos en los que funcionará el sistema, los cuales comienzan a desarrollarse en el flujo de trabajo de diseño y que se perfeccionan en la implementación (Vázquez Ventura, y otros, 2013).

#### 3.3.1. Diagrama de despliegue.

El diagrama de despliegue representa la arquitectura de tiempo de ejecución de los procesadores, dispositivos y los componentes de software que se ejecutan en esa arquitectura. Es una descripción física de la topología del sistema y describe la estructura de las unidades de hardware (Mediavilla, 2004). Además, representa la disposición de las instancias de componentes de ejecución en instancias de nodos conectados por enlaces de comunicación. Un nodo es un recurso de ejecución tal como un procesador, un

dispositivo o memoria. En los procesadores es donde se encuentran alojados los componentes (Prezi Inc, 2015). La Figura 8 que se muestra a continuación representa el diagrama de despliegue del Sistema de Gestión de la Información para el control del Plan y Presupuesto y está compuesto por:

- **PC\_Cliente:** Representa una computadora desde la cual el usuario podrá acceder a la aplicación.
- **Impresora:** Para que el usuario imprima los modelos y reportes, está conectado con la PC\_Cliente por puerto USB.
- **Servidor web (Apache):** Establece comunicación con la PC\_Cliente utilizando el protocolo HTTPS y realiza peticiones mediante TCP/IP al servidor de Bases de Datos.
- **Servidor de base de datos:** Representa el servidor donde estará el SGBD que da respuesta a las peticiones realizadas por la aplicación.
- **Servidor de servicio de autenticación:** Establece comunicación con el servidor web utilizando el protocolo HTTPS y posibilita la autenticación para el uso de la aplicación utilizando el protocolo LDAP.

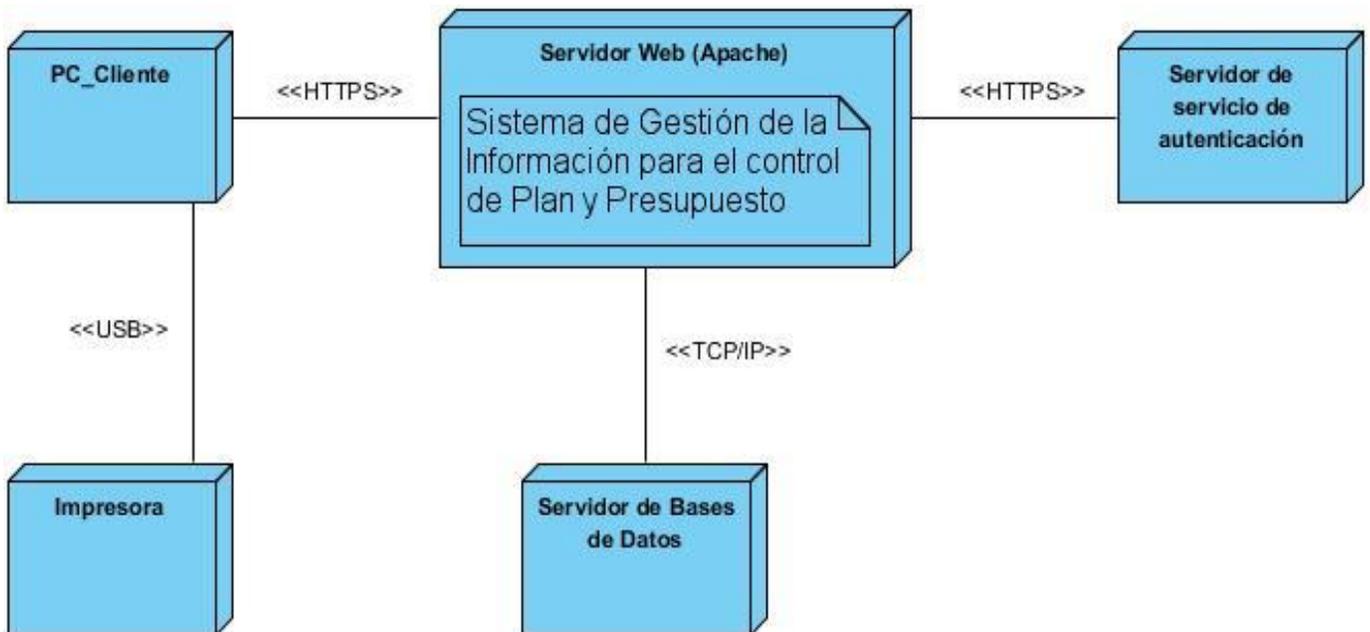
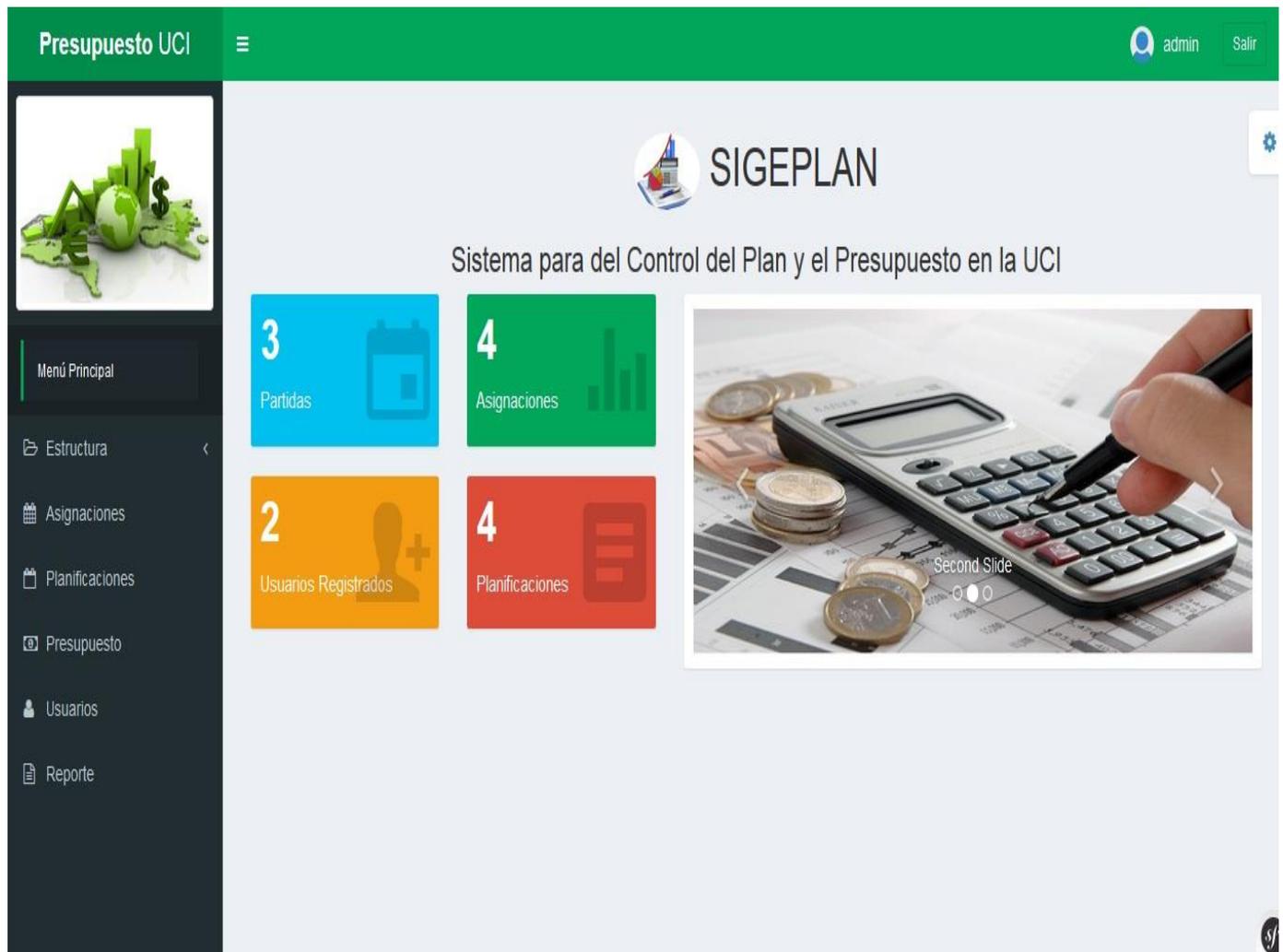


Figura 8: Diagrama de despliegue.

### 3.4. Pantallas principales de la aplicación.

A continuación se presentan las principales interfaces del Sistema de Gestión de la Información para el control del Plan y Presupuesto.

La Figura 9 muestra la interfaz principal del sistema, esta da paso a las principales funcionalidades del sistema como: Gestionar datos de partidas, Gestionar asignación y Exportar reportes a formato xls.



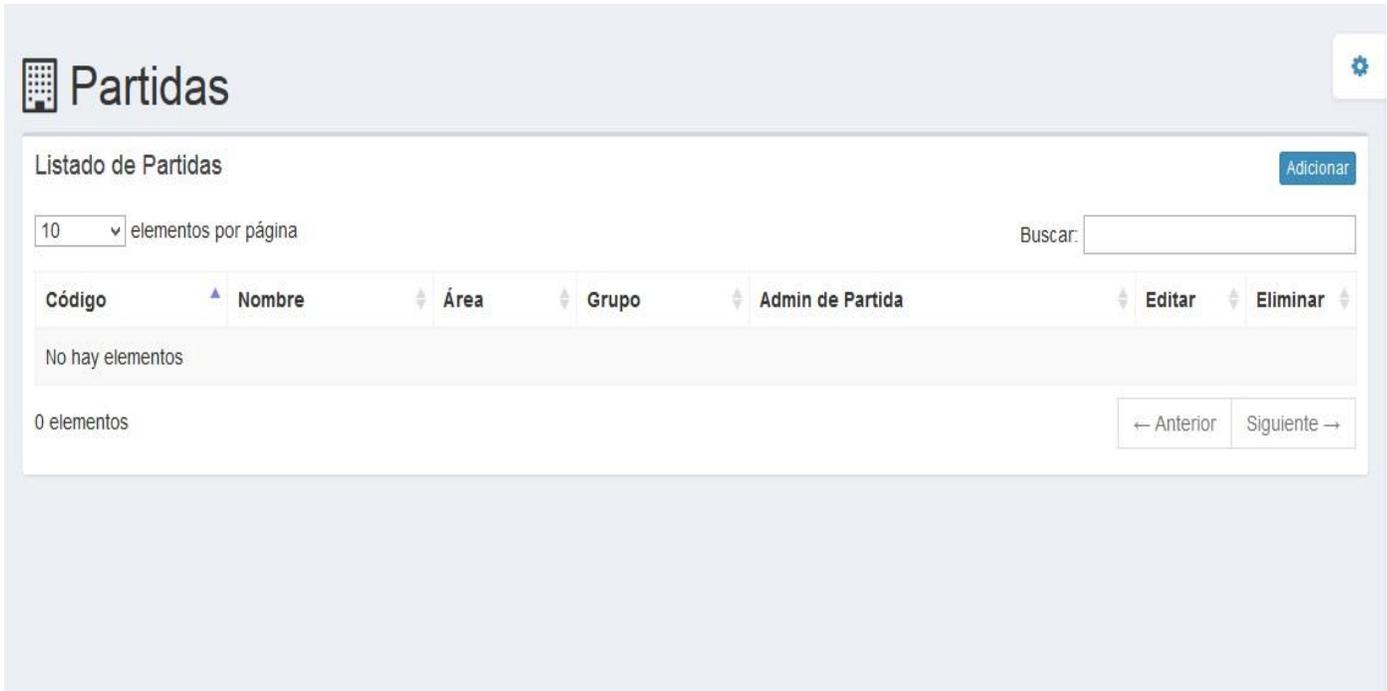
**Figura 9: Interfaz principal del sistema.**

La Figura 10 muestra la interfaz de autenticación del usuario en el sistema, la cual da paso a la Figura anterior.



**Figura 10: Interfaz de inicio del sistema.**

La Figura 11 muestra la funcionalidad gestionar datos de partida.



**Figura 11: Interfaz de la funcionalidad gestionar datos de partidas.**

### **3.5. Validación de la solución propuesta.**

Las pruebas de software existen debido a la capacidad natural de provocar errores. Una vez generado el código fuente, el software debe ser probado para descubrir y corregir el máximo de errores posibles antes de su entrega al cliente. El objetivo es diseñar una serie de casos de prueba que tengan una alta probabilidad de encontrar errores. Aquí es donde se aplican las técnicas de pruebas del software (Pressman Roger S., 2001).

Para la validación y verificación de SIGEPLAN se realizaron pruebas unitarias y de aceptación. Además se validaron las variables del problema de la investigación.

#### **3.5.1. Pruebas unitarias.**

La metodología XP se basa en pruebas unitarias para comprobar el funcionamiento de los códigos que se vayan implementando. Las pruebas unitarias se utilizan para detectar errores en el código, pues se centran en la menor unidad de diseño del software (módulos, método de una clase, clases). Tiene como objetivo principal detectar errores en los principales procesos al ser ejecutado de manera independiente. Estas son diseñadas por los programadores y realizadas ante cualquier modificación del sistema para detectar fallas que pudieran ocurrir (Ramos Roman, y otros, 2007).

Las pruebas de unitarias proporcionan algunas ventajas como que impulsa al desarrollador a crear una estructura de código más simple (lo que se conoce como refactorización). Simplifica la integración, pues permite llegar a la fase de integración con un código que funcione correctamente y facilita un mejor entendimiento del código ya que lo documenta a través de las pruebas (Córdova Dorta, 2011). Estas siempre están orientadas a las técnicas de caja blanca.

- **Pruebas de Caja Blanca.**

Las pruebas de caja blanca, también conocidas como pruebas de caja transparente o pruebas estructurales, se centran en los detalles procedimentales del software, por lo que su diseño está fuertemente ligado al código fuente. Este método intenta garantizar que se ejecutan al menos una vez todos los caminos independientes que presente el módulo y que todas las estructuras de datos internas serán usadas (Pressman Roger S., 2001).

Para el sistema desarrollado es necesario evidenciar si es un producto de alta calidad, por tanto es preciso valorar que tan certera ha sido la implementación. Por tanto se debe aplicar una de las técnicas que comprende las pruebas de caja blanca, en este caso la del camino básico o ruta básica. Para ello es necesario conocer el número de caminos independientes, por lo que se construirá un grafo asociado y se calculará de la complejidad ciclomática.

Un Grafo de Flujo está formado por 3 componentes fundamentales que ayudan a su elaboración, comprensión y brinda información para corroborar que el trabajo se está haciendo adecuadamente. Los componentes son:

- **Nodo:** Cada círculo representado se denomina nodo del Grafo de Flujo, el cual representa una o más secuencias procedimentales. Un solo nodo puede corresponder a una secuencia de procesos o a una sentencia de decisión. Puede ser también que hallan nodos que no se asocien, se utilizan principalmente al inicio y final del grafo.
- **Flechas:** Las flechas del grafo se denominan aristas y representan el flujo de control. Una arista debe terminar en un nodo, incluso aunque el nodo no represente ninguna sentencia procedimental.
- **Regiones:** Las regiones son las áreas delimitadas por las aristas y nodos. También se incluye el área exterior del grafo, contando como una región más. Las regiones se enumeran y la cantidad de regiones es equivalente a la cantidad de caminos independientes del conjunto básico de un programa.

A continuación se analiza el código y se enumeran las sentencias de código del procedimiento realizado sobre el método *createAction (Request \$request)* encargado de adicionar una partida en el sistema, ver Figura 12 y el grafo de flujo asociado al mismo Figura 13.

```

45 public function createAction(Request $request)
46 {
47     $entity = new Partida();
48     $form = $this->createCreateForm($entity);
49     $form->handleRequest($request);
50 }
51 if ($form->isValid()) {
52     $em = $this->getDoctrine()->getManager();
53     $em->persist($entity);
54     $em->flush();
55 }
56 return $this->redirect($this->generateUrl('partida'));
57 }
58
59 return array(
60     'entity' => $entity,
61     'form'   => $form->createView(),
62 );
63 }

```

Figura 12: Método createAction(Request \$request).

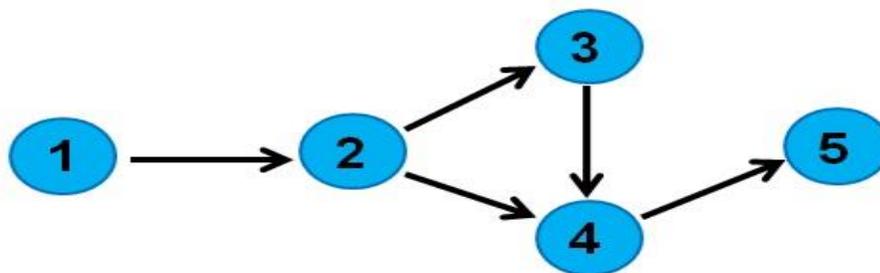


Figura 13: Grafo del flujo asociado al método adicionar partida.

Fórmulas para calcular la complejidad ciclomática:

1.  $V(G) = (A - N) + 2$

Donde “A” es la cantidad de aristas y “N” es la cantidad de nodos.

$$V(G) = (5 - 5) + 2$$

$$V(G) = 2$$

### 2. $V(G) = P + 1$

Siendo “P” la cantidad de nodos predicados (son los nodos de los cuales parten dos o más aristas).

$$V(G) = 1 + 1$$

$$V(G) = 2$$

### 3. $V(G) = R$

Donde “R” representa la cantidad de regiones en el grafo.

$$V(G) = 2$$

El cálculo efectuado mediante las fórmulas ha dado el mismo valor, por lo que se puede decir que la complejidad ciclomática del código es 2, lo que significa que existen dos posibles caminos por donde el flujo puede circular, este valor representa el límite mínimo del número total de casos de pruebas para el procedimiento tratado.

| Número | Caminos básicos |
|--------|-----------------|
| 1      | 1-2-3-4-5       |
| 2      | 1-2-4-5         |

**Tabla 8: Caminos básicos del flujo asociado al método adicionar partida.**

| Caso de prueba   |
|--|
| <b>Camino:</b> 1-2-3-4-5   |
| <b>Entrada:</b> Recibe número positivo, se introduce nombre y se selecciona un grupo, área y administradores.                    |
| <b>Resultado:</b> Se introduce satisfactoriamente.   |
| <b>Condiciones:</b> Tiene que estar registrado al menos un grupo y un usuario. Además el nombre debe tener al menos un apellido. |

**Tabla 9: Caso de prueba unitaria.**

| Caso de prueba |
|----------------|
|----------------|

|  |
|--|
| <b>Camino:</b> 1-2-4-5   |
| <b>Entrada:</b> Recibe número negativo, se introduce nombre y se selecciona un grupo, área y administradores.                    |
| <b>Resultado:</b> Muestra un mensaje notificando el error.   |
| <b>Condiciones:</b> Tiene que estar registrado al menos un grupo y un usuario. Además el nombre debe tener al menos un apellido. |

**Tabla 10: Caso de prueba unitaria.**

### 3.5.2. Pruebas de aceptación.

Las pruebas de aceptación o pruebas funcionales como también son conocidas, son las pruebas finales antes de la liberación del sistema (Letelier, y otros, 2006). Se realizan con el objetivo de confirmar si el software está listo y puede ser usado por usuarios finales para ejecutar aquellas funciones y tareas para las cuales fue implementado (Machado Peña, y otros, 2012). En XP las pruebas de aceptación son creadas a través de las HU. El cliente o usuario detalla los diversos aspectos a comprobar para verificar que una HU ha sido correctamente implementada. Las HU pueden tener varias pruebas de aceptación, para garantizar su buen funcionamiento. Las pruebas de aceptación son consideradas como “pruebas de caja negra”, las cuales están orientadas a los requisitos funcionales del software, buscan que se hayan ingresado toda clase de entrada y que la salida obtenida es igual a la esperada (Pressman, 2005).

A continuación se muestra algunas de las pruebas realizadas a la HU; que constituye una de las funcionalidades primordiales para el cliente, las restantes se encuentran en el Anexo 5.

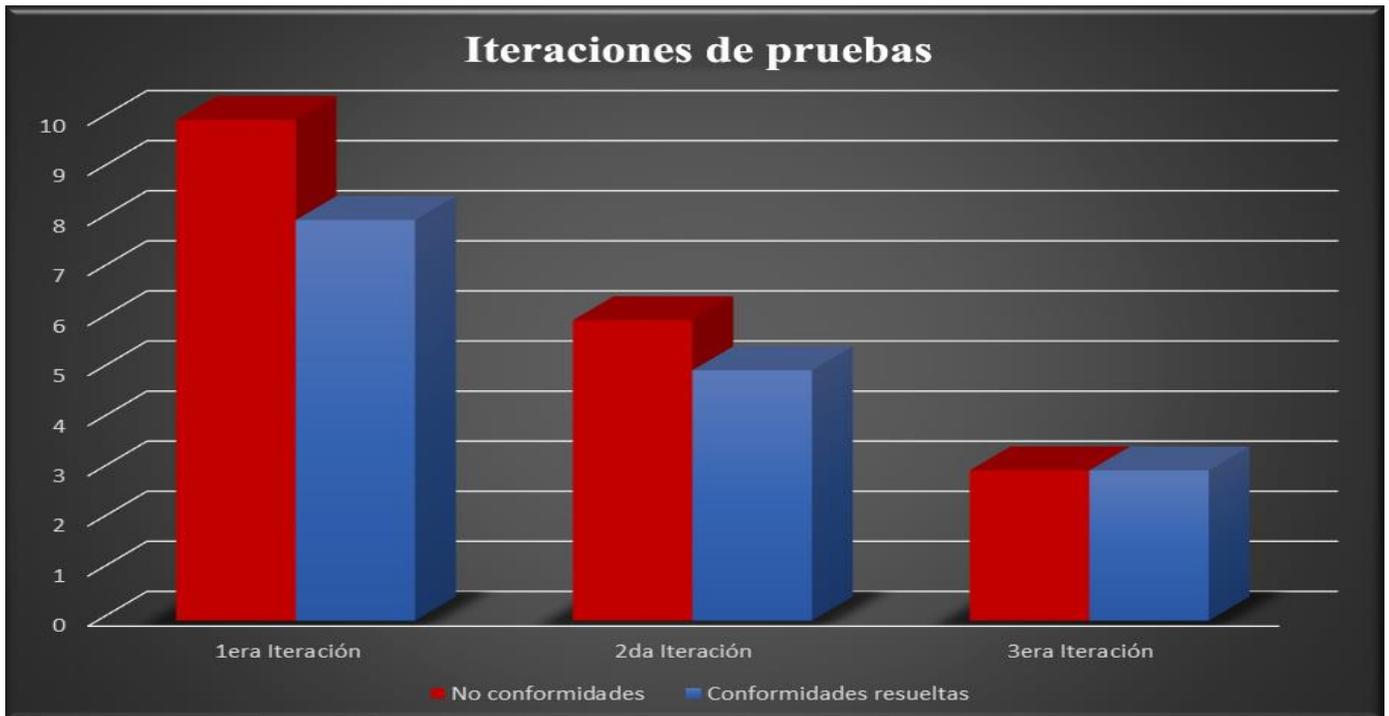
| <b>Adicionar datos de partidas</b>                    |  |   |  |
|---|--|---|--|
| <b>Escenario</b>                                      | <b>Descripción</b>   | <b>Respuesta del sistema</b>  | <b>Flujo Central</b>   |
| EC 1.1 Agregar datos de partidas de forma correcta.   | El sistema agrega los datos de las partidas correctamente.   | El sistema almacena la información y muestra un listado de las partidas creadas.  | El usuario autenticado en el sistema con el rol de Especialista, accede al módulo “ <b>Estructura</b> ”, ubicado en la parte superior izquierda del sistema.   |
| EC 1.2 Agregar datos de partidas de forma incorrecta. | El sistema agrega los datos de partidas de forma incorrecta. | El sistema no agrega los datos de las partidas y muestra los siguientes mensajes de error:<br>Para el campo código: “Valor incorrecto, debe ser número.”<br>Para el campo nombre: “El valor introducido no es válido.”<br>En el caso de que exista una partida ya creada con datos iguales: “Existe una partida con este código.” | 1.- El sistema le muestra al usuario un listado de las partidas ya creadas en él.<br>2.- El usuario realiza un clic en el botón “ <b>Adicionar</b> ”.<br>3.- El sistema muestra un formulario para que el usuario introduzca los datos.<br>4.- El usuario introduce la información y presiona el botón “ <b>Adicionar</b> ”. |

**Tabla 11: Caso de prueba. Adicionar datos de partidas.**

| <b>Editar datos de partidas</b>                      |   |  |  |
|--|---|--|--|
| <b>Escenario</b>                                     | <b>Descripción</b>  | <b>Respuesta del sistema</b>   | <b>Flujo Central</b>   |
| EC 1.1 Editar datos de partidas de forma correcta.   | El sistema edita los datos de las partidas correctamente.   | El sistema almacena la información cambiada y muestra un listado de las partidas creadas.  | El usuario autenticado en el sistema con el rol de Especialista, accede al módulo <b>"Estructura"</b> , ubicado en la parte superior izquierda del sistema.  |
| EC 1.2 Editar datos de partidas de forma incorrecta. | El sistema edita los datos de partidas de forma incorrecta. | El sistema no edita los datos de las partidas y muestra los siguientes mensajes de error:<br>Para el campo código: "Valor incorrecto, debe ser número."<br>Para el campo nombre: "El valor introducido no es válido."<br>En el caso de que exista una partida ya creada con datos iguales: "Existe una partida con este código." | 1.- El sistema le muestra al usuario un listado de las partidas ya creadas en él, con las opciones de <b>"Editar"</b> y <b>"Eliminar"</b> .<br>2.- El usuario realiza un clic en la opción <b>"Editar"</b> .<br>3.- El usuario accede la opción: <b>"Editar"</b> y el sistema muestra un formulario con los siguientes campos:<br>• Código (obligatorio, campo numérico)<br>• Nombre (obligatorio, campo de texto)<br>• Área (estas viene predefinidas en un combobox).<br>• Grupo (esta opción viene predefinida en un combobox).<br>El usuario modifica la información y presiona el botón: <b>"Aceptar"</b> . |

**Tabla 12: Caso de prueba. Editar datos de partidas.**

Durante el desarrollo del sistema se realizaron pruebas de aceptación a las HU desarrolladas por cada iteración, verificando que las funcionalidades que estas contienen, se desempeñen correctamente. El gráfico que se muestra a continuación establece una relación entre las iteraciones de pruebas realizadas y el número de no conformidades detectadas en las mismas.



**Figura 14: Resultado de las pruebas.**

A partir de los casos de pruebas definidos por el cliente durante tres iteraciones de pruebas, se detectaron en cada una de estas 10, 6 y 3 no conformidades respectivamente, llegando al total de 16 no conformidades, cabe añadir que de la primera iteración quedaron dos no conformidades sin resolver las cuales se adicionaron a la segunda iteración y en esta iteración quedó una, la cual se adicionó a la tercera. Las pruebas realizadas revelaban errores en la documentación, codificación y en el diseño de las interfaces de usuario que en la mayoría de los casos no incidían significativamente en las respuestas esperadas. Todas estas dificultades fueron solucionadas en el menor tiempo posible, siempre antes de pasar a la iteración siguiente. Una vez llegada a la solución final, se le demostró al cliente el cumplimiento de todo los requisitos solicitados.

### **3.5.3. Validación de las variables de la investigación.**

La validación de las variables de la investigación se realizó con el propósito de evaluar que la solución propuesta mejora el proceso de Elaboración de Plan y Presupuesto en la UCI, de forma tal que se garantice el control y se agilice el procesamiento de la información.

- **Agilizar el procesamiento de la información:**

Los administradores de partidas elaboran sus propuestas del Plan y las entregan en el área de Planificación y Estadísticas en un dispositivo de almacenamiento. Los especialistas de esta área deben conformar cada uno de los modelos emitidos por los administradores de partidas para la Elaboración del Plan y Presupuesto. Posteriormente, se designan especialistas del área para recibir el CUP y CUC. Seguidamente se realiza la captación, la revisión y la introducción de los datos en los modelos correspondientes. Este proceso demora aproximadamente de 45 días.

Con el despliegue del sistema se agiliza el proceso debido a:

1. Los administradores de partidas introducen los importes de las partidas en el sistema y no es obligatorio presentarse personalmente ante el especialista del área.
2. Una vez aceptados los datos introducidos por los administradores de partida se reduce considerablemente el tiempo del proceso, debido a que los especialistas del área no tienen que introducir los datos en los modelos del Plan y Presupuesto, puesto que se generan automáticamente a través de los reportes correspondientes.

- **Control:**

La Dirección de Planificación y Estadísticas lleva el control del proceso que se realiza para la propuesta de Plan y Presupuesto. Las especialistas de esta área conocen el importe de las partidas que proponen cada uno de los administradores y además interactúan con estos en caso de inconformidad con la solicitud propuesta, mediante una notificación. De ser necesario se reúnen personalmente, de esta forma se demuestra el intercambio que existe entre la Dirección y sus administradores.

Con el despliegue del sistema se garantiza el control debido a:

1. Los administradores de partida insertan el importe en el sistema garantizando la seguridad de la información debido a que esta no se tiene que transportar mediante dispositivos de almacenamiento.
2. Los especialistas pueden chequear constantemente la información implícita en el sistema.

### **3.6. Conclusiones Parciales.**

El uso de los estándares de codificación definidos por el framework Symfony2 permitió desarrollar un código reutilizable y simple. La representación de los distintos diagramas expuestos y las pruebas creadas, visualizaron la estructura y el funcionamiento del sistema. Se llevó a cabo la ejecución de varias iteraciones, teniendo como respuesta varias no conformidades del sistema, las cuales fueron solucionadas en poco tiempo, con el objetivo de validar su correcto funcionamiento.

## **CONCLUSIONES GENERALES**

- El estudio de sistemas de gestión presupuestaria permitió comprender y aplicar las teorías existentes al sistema propuesto en la investigación.
- El análisis del negocio y las entrevistas realizadas al personal del área de Planificación y Estadística, permitieron elaborar los requisitos funcionales y no funcionales del sistema.
- La elección correcta de las herramientas a utilizar ayudó a garantizar un desarrollo ágil del sistema.
- La implementación de las funcionalidades contribuyó a la gestión de la información que maneja la Elaboración del Plan y Presupuesto en la UCI.
- Mediante la aplicación de pruebas de software y un aval del cliente quedó validado el sistema.

## **RECOMENDACIONES**

- Capacitar el personal encargado de interactuar con la aplicación, con cursos o seminarios.
- Realizar salvallas constantes de la información.
- Ampliar el sistema desarrollado para el área de Planificación y Estadística, hasta lograr cubrir las necesidades del sistema utilizado por la misma, para la gestión financiera.
- Por la utilidad práctica del sistema, se considera que puede ser usado por otras Universidades o entidades del país.

## BIBLIOGRAFÍA

**Arredondo Gamboa, Keiry. 2012.** *Servicio Web 2.0 Apache vs IIS.* 2012.

**Benghazi, Kawtar y Garrido Bullejos, José Luis. 2010.** *Introducción al Modelado de Procesos del Negocio.* Granada : s.n., 2010.

**Bizagi. 2012.** Bizagi. [En línea] 2012. [Citado el: 10 de marzo de 2015.] <http://www.buenastareas.com/ensayos/Bizagi/4235294.html>.

**Bootstrap. 2013.** *Bootstrap.* 2013.

**Burbani, Ruiz Jorge E. 2005.** *Presupuestos: enfoque de gestión, planeación y control de recursos.* Bogotá : s.n., 2005.

**CASA CONSULTORA DISAIC. 2015.** *EL CONSULTOR ELECTRONICO DEL CONTADOR Y EL AUDITOR.* La Habana : s.n., 2015.

**Cervantes, Humberto. 2010.** SG Buzz Conocimiento para Crear Software Grandioso. [En línea] 27 de febrero de 2010. [Citado el: 2 de abril de 2015.] <http://sg.com.mx/revista/27/arquitectura-software>.

**CIPIMM. 2015.** Centro de Investigaciones para la Industria Minero-Metalúrgica. *Centro de Investigaciones para la Industria Minero-Metalúrgica.* [En línea] 2015. [Citado el: 8 de marzo de 2015.] <http://www.cipimm.co.cu/sitio-info/SISCONT5.php>.

**Colina Luisa. 2012.** *La Planificación Presupuestaria en Universidades Públicas.* Universidad de Zulia, Venezuela : s.n., 2012.

**Córdova Dorta, Erick. 2011.** *Procedimiento para la ejecución de las pruebas de unidad a los componentes de la Capa de presentación del sistema CEDRUX.* La Habana : Universidad de las Ciencias Informáticas, 2011.

**Definición. 2009.** Definición de. [En línea] 2009. <http://definicion.de/plan/#ixzz3LPZhTJHc>.

**Diego, Vallespir. 2002.** *CRC y un Taller.* Montevideo, Uruguay : s.n., 2002.

**Doyle, Matt. 2012.** *Beginning PHP 5.3.* 2012.

eco-finanzas. *Diccionario Economía - Administración - Finanzas - Marketing.* [En línea] [Citado el: 16 de febrero de 2015.] [http://www.eco-finanzas.com/diccionario/P/PARTIDAS\\_DE\\_GASTO.htm](http://www.eco-finanzas.com/diccionario/P/PARTIDAS_DE_GASTO.htm).

**Ecured. 2015.** Ecured. *Ecured.* [En línea] 2015. [http://www.ecured.cu/index.php/Gesti%C3%B3n\\_de\\_la\\_Informaci%C3%B3n](http://www.ecured.cu/index.php/Gesti%C3%B3n_de_la_Informaci%C3%B3n).

—. 2015. Ecured. [En línea] 2015. [Citado el: 21 de abril de 2015.] [http://www.ecured.cu/index.php/Patrones\\_en\\_Symfony](http://www.ecured.cu/index.php/Patrones_en_Symfony).

**eTES. 2015.** eTES Gestión Empresarial. [En línea] 2015. [Citado el: 17 de febrero de 2015.] <http://www.datys.cu/wpinfo/producto.aspx?8..>

**Ferreira dos Santos, Joao Carlos. 2012.** *Criando Diagramas UML com o StarUML.* 2012.

**Gavilán, Fran. 2009.** Internet Information Services (IIS). *Internet Information Services (IIS).* [En línea] 17 de noviembre de 2009. [Citado el: 6 de febrero de 2015.] <http://terrafx9.blogspot.com/2011/11/internet-information-services-iis.html>.

**Gómez Soriano, José Manuel. 2014.** *Motores MySQL.* 2014.

**González San Segundo, Eva María. 2013.** *Proyecto complejo deportivo.* s.l. : Universidad Oberta de Catalunya, 2013.

**Hernández, Rolando, León y Coello, Alfredo y González, Sayda. 2011.** *El Proceso de Investigación Científica.* La Habana : Editorial Universitaria, 2011.

**Howard, Yvonne. 2008.** EdShare. [En línea] 2008. [Citado el: 25 de febrero de 2015.] <http://www.edshare.soton.ac.uk/11845>.

**Hristov, Alexander. 2007.** *Manual de Estilo de Programación.* s.l. : PLANETALIA Formación y Consultoría, 2007.

**Iglesia de Scientology Internacional. 2011.** Scientology . *Scientology* . [En línea] 2011. [http://spanish.scientologyhandbook.org/sh17\\_3.htm](http://spanish.scientologyhandbook.org/sh17_3.htm).

**Informáticas, Universidad de las Ciencias. 2013.** Entorno Virtual de Aprendizaje. [En línea] 2013. [Citado el: 15 de febrero de 2015.] [http://eva.uci.cu/file.php/161/Documentos/Materiales\\_complementarios/UD\\_1\\_Procesos/Metodologias/Metodologias\\_agiles\\_en\\_el\\_desarrollo\\_de\\_software.pdf](http://eva.uci.cu/file.php/161/Documentos/Materiales_complementarios/UD_1_Procesos/Metodologias/Metodologias_agiles_en_el_desarrollo_de_software.pdf).

**Jácome, Santiago, Tituaña Cumbal, Walter Celiano y Torres Cañizares, Edwin Jesús. 2009.** *Elaboración de un manual de la plataforma Netbeans Ide para la Disicom.* s.l. : LATACUNGA / ESPE / 2009, 2009.

**Jesús Granda, Ana, Benito, De y Aragón y Sánchez, Rodolfo. 2007.** *Manual de Presto 4ta Edición.* Madrid. España : McGraw-Hill/Interamericana de España S.A., 2007. ISBN: 9788448156206.

**Leiseca, Hernández Esteban. 2010.** *Sistema de Planificación del Presupuesto del Departamento de Transporte de la Unidad de Aseguramiento Aduanal.* Ciudad de La habana : s.n., 2010.

**Letelier, Patricio y Penadés, Carmen. 2006.** *Métodologías ágiles para el desarrollo de software: eXtreme Programming (XP).* Valencia: Universidad Politécnica : s.n., 2006.

- Machado Peña, Yadira, y otros. 2012.** *Estrategia de Pruebas Funcionales Aplicada en Calisoft para Garantizar la Calidad del Software*. La Habana: Universidad de las Ciencias Informáticas : s.n., 2012.
- Marciszack, Marcelo. 2010.** *Modelos de especificación de requerimientos para la obtención de esquemas conceptuales en un dominio restringido: comparación de metodologías*. 2010.
- Marcos Delgado, Carlos Fco; Poyo Garrido, Alejandro. 2013.** *COMPARATIVA FRAMEWORKS, Desarrollo de Aplicaciones Web II*. 2013.
- Martín, Eduardo Luis. 2014.** Un efectivo sistema cubano de contabilidad. [En línea] 2014. [Citado el: 16 de diciembre de 2014.] <http://www.eleconomista.cubaweb.cu/2005/edicionimpresa/premioeconomiamiguel.html>.
- Martinez, R.L. 2015.** Openstax. [En línea] 2015. [Citado el: 2 de marzo de 2015.] [http://cnx.org/contents/8c4c8090-9bbf-4ae1-883f-b875915d716b@3/Java4310:\\_Getting\\_Started\\_with](http://cnx.org/contents/8c4c8090-9bbf-4ae1-883f-b875915d716b@3/Java4310:_Getting_Started_with).
- Mediavilla, Elena. 2004.** *Máster de Computación. Programación Orientada a Objetos*. 2004.
- 2014.** Metodología OpenUp. *Metodología OpenUp*. [En línea] 2014. <http://openup3.blogspot.com/2014/02/metodologia-open-up.html>.
- MFP, Ministerio de Finanzas y Precios. 2014.** *CLASIFICADOR POR OBJETO DEL GASTO*. La Habana : s.n., 2014.
- Naramore, Elizabeth. 2005.** *Beginning PHP5, Apache, and MySQL Web Development*. 2005.
- Novella Latorre, Javier. 2012.** *Sistema de gestión de base de datos PostgreSQL*. 2012.
- Ordaz Zubia, Velia y Saldaña García, Gloria. 2006.** *ANÁLISIS Y CRÍTICA DE LA METODOLOGÍA PARA LA REALIZACIÓN DE PLANES REGIONALES EN EL ESTADO DE GUANAJUATO*. Guanajuato : s.n., 2006. ISBN: 84-689-9176-7.
- Orjuela, A. 2008.** *The methodologies of Agil Development like and Opportunity for the Inginneering of Educative Software*. 2008.
- Oswaldo, Castillo, Daniel, Figueroa y Hector, Sevilla. 2014.** Programación Extrema. [En línea] 2014. [Citado el: abril de 15 de 2015.] <http://programacionextrema.tripod.com/index.htm>.
- Parral, César Augusto. 2014.** Portal de la INVESTIGACIÓN. [En línea] 2014. [Citado el: 11 de marzo de 2015.] [http://www.vinv.ucr.ac.cr/index.php?option=com\\_content&view=article&id=1881:ucr-estrena-nuevo-sistema-de-formulacion-de-proyectos&catid=1&Itemid=68](http://www.vinv.ucr.ac.cr/index.php?option=com_content&view=article&id=1881:ucr-estrena-nuevo-sistema-de-formulacion-de-proyectos&catid=1&Itemid=68).
- Potencier, Fabien y Zaninotto, François. 2010.** *La Guía Definitiva*. 2010.

- Pressman, Roger S. 2005.** *Ingeniería de Software. Un enfoque práctico. Sexta Edición.* España: McGraw Hill : s.n., 2005.
- . **2001.** *Ingeniería del Software. Un enfoque práctico. Quinta Edición.* España: McGraw Hill : s.n., 2001.
- Prezi Inc. 2015.** PREZI. [En línea] 2015. [Citado el: 9 de mayo de 2015.] <https://prezi.com/4yhb-pni-k/diagrama-de-despliegue/>.
- Ramos Roman, Isabel, Tuya, Javier y Dolado Cosín, José Javier. 2007.** *Técnicas cuantitativas para la gestión de la ingeniería del software.* Gesciblo : s.n., 2007.
- Rodríguez Borrego, Leyner. 2010.** *Sistema de Adquisición de datos para Mediciones Multiconto.* Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas. La Habana: Universidad de las Ciencias Informáticas : s.n., 2010.
- Scribd. Inc. 2015.** Scribd. [En línea] 2015. [Citado el: 5 de marzo de 2015.] <http://es.scribd.com/doc/12227873/Concepto-y-Caracteristicas-Del-Sistema-de-Gestion-de-Informacion#scribd>.
- Softland Inversiones S.L. 2015.** Softland ERP. [En línea] 2015. [Citado el: 16 de febrero de 2015.] <http://www.softland.cr/softland-erp..>
- Sommerville, Ian. 2002.** *Ingeniería de Software. Sexta Edición.* Prentice-Hall : s.n., 2002. ISBN 970-26-0206-8.
- Torres Almira, Liuba. 2012.** *Intranet del Ministerio de Informática y las Telecomunicaciones.* Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas. La Habana: Universidad de las Ciencias Informáticas : s.n., 2012.
- Vázquez Chacón, Elisa. 2013.** *Plugin generador de StoryBoards para el Sistema Gestor de Procesos de Medía.* Universidad de las Ciencias Informáticas, La Habana : s.n., 2013.
- Vázquez Ventura, Merlyn y Rodríguez Quintana, Daniel Raúl. 2013.** *Sistema de gestión de información de las redes sociales Facebook y Twitter.* Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas. La Habana: Universidad de las Ciencias Informáticas : s.n., 2013.
- Villalón, Mariela. 2011.** *La Planificación y el Modelo Económico Cubano.* Santiago de Cuba : s.n., 2011. ISSN 2218-3639.

## **GLOSARIO DE TÉRMINOS**

**LDAP:** Es un protocolo a nivel de aplicación, que permite el acceso a un servicio de directorio ordenado y distribuido para buscar diversas informaciones en el entorno de red. LDAP también es considerado una base de datos (aunque su sistema de almacenamiento puede ser diferente) al que pueden realizarse consultas. Habitualmente, almacena la información de login (usuario y contraseña) y es utilizado para autenticarse aunque es posible almacenar otra información (datos de contactos del usuario, ubicación de diversos recursos de la red, permisos, certificados, etc.).

**Multiplataforma:** Término utilizado frecuentemente en informática para indicar la capacidad o características de poder funcionar o mantener una interoperabilidad de forma similar en diferentes sistemas operativos o plataformas.