

**FACULTAD DE INFORMÁTICA Y MATEMÁTICA
DEPARTAMENTO DE INGENIERÍA INFORMÁTICA**

Trabajo de Diploma

En opción al título de Ingeniero Informático

**Sistema de apoyo a la gestión de la información del Plan de la
Economía de la Universidad de Holguín
“Oscar Lucero Moya”**

Autor: Ernesto Arias Araus

Tutores: M.Sc. Ana de Lourdes Torralbas Blázquez
Lic. Lorenzo Luis Peña Marrero

**Holguín
Curso 2013-2014**

Declaración de Autoría

Declaro que soy el único autor de este trabajo y autorizo al Departamento de Informática de la Universidad de Holguín “Oscar Lucero Moya” para que hagan el uso que estimen pertinente con este trabajo.

Para que así conste firmo la presente a los _____ días del mes de _____ del _____.

Diplomante: _____

Ernesto Arias Arauz

M.Sc. Ana de Lourdes Torralbas Blázquez

Lic. Lorenzo Luis Peña Marrero

Opinión del usuario

El Trabajo de Diploma, titulado _____,
fue realizado en nuestra entidad _____. Se
considera que, en correspondencia con los objetivos trazados, el trabajo
realizado le satisface:

- Totalmente
- Parcialmente en un ____ %

Los resultados de este Trabajo de Diploma le reportan a esta entidad los
beneficios siguientes:

Como resultado de la implantación de este trabajo se reporta un efecto
económico que asciende a _____ MN y/o _____ CUC.

Y para que así conste, se firma la presente a los ____ días del mes de
_____ del año _____.

Nombre del representante de la entidad: _____

Cargo: _____

Firma y cuño

Opinión del tutor

Título: _____

Autor: _____

El tutor del presente Trabajo de Diploma considera que durante su ejecución el estudiante mostró las cualidades que a continuación se detallan.

Independencia

Originalidad

Creatividad

Laboriosidad

Responsabilidad

Por todo lo anteriormente expresado considero que el estudiante está apto para ejercer como Ingeniero Informático; y propongo que se le otorgue al Trabajo de Diploma la calificación de _____.

M.Sc. Ana de Lourdes Torralbas Blázquez

Lic. Lorenzo Luis Peña Marrero

Fecha: _____

RESUMEN

En la Vicerrectoría económica de la Universidad de Holguín “Oscar Lucero Moya” se lleva a cabo la planificación del presupuesto que financia los procesos internos de la entidad. Este presupuesto consta de una parte en moneda nacional y otra en divisas (CUC). Las tareas de planificación y administración del presupuesto en divisas se caracterizan por su complejidad, ya que las mismas requieren una cuantiosa documentación y se realizan de forma manual. Por ello que surge la necesidad de crear un sistema que permita la automatización de este proceso, teniendo en cuenta que ya existe uno para la gestión del presupuesto en moneda nacional.

La presente investigación pretende resolver esta deficiencia, a partir del desarrollo de una aplicación informática que permita gestionar toda la información referente al presupuesto en divisas de forma eficiente, que proporcione seguridad e integridad de los datos, así como la actualización de los mismos, para que dicho proceso se pueda desarrollar con calidad en la Universidad de Holguín "Oscar Lucero Moya" y facilite el trabajo del personal que lleva a cabo su control.

Para el desarrollo de esta aplicación se empleó la metodología de desarrollo de software ICONIX, con la herramienta de modelado *Enterprise Architect*, como gestor de bases de datos MySQL, el lenguaje de programación Python y el Framework para aplicaciones web Django, permitiendo obtener un producto final sostenible que cumple con los requerimientos especificados.

ABSTRACT

In the economical vice-rectory of the University of Holguín “Oscar Lucero Moya” takes place the process of planning the budget that supports the internal process of the entity. This budget has one component in our national currency and another in CUC currency. The task of planning and administration of the budget in CUC are specially complicated and generate a huge paperwork because are made manually, that’s why arises the necessity of creating a system that allows to automate this process, taking into account that already exists a system to manage the budget in our national currency.

The present investigation is intended to solve this deficiency, having as start point the implementation of an application web that allows managing the budget in the CUC currency, which will guarantee the efficiency of the process, the safety and integrity, and the actualization of the data. All these will make easier the work of all the people that have to do this process manually.

In order to develop the system were used free tools and technologies, such as the software development methodology ICONIX, with the modeling tool Enterprise Architecture, as the data base manager was used MySQL, the programming language for web pages Python and the framework Django for developing the web part of the application. It allows obtaining a final sustainable product that follows the specified requirements.

Índice

Introducción	1
CAPÍTULO 1: FUNDAMENTOS TEÓRICOS	6
1.1. Breve acercamiento al análisis económico-financiero.....	6
1.2. Descripción de la entidad.....	7
1.3. Conceptos fundamentales	8
1.4. Sistemas Gestores de Información	9
1.5. Software de gestión de información en el control económico-financiero.....	10
1.6. Tendencias y tecnologías actuales.....	12
Conclusiones parciales del capítulo.....	27
CAPÍTULO 2. DESCRIPCIÓN DE LA SOLUCIÓN.....	28
2.1. Modelo del dominio	28
2.2. Modelación de los casos de uso.....	33
2.3. Análisis y diseño preliminar.....	37
2.4. Diseño detallado.....	44
2.5. Implementación	49
2.6. Prueba.....	49
2.7. Valoración de sostenibilidad.....	50
2.8. Valoración de la Propuesta de Solución	53
Conclusiones parciales del Capítulo	55
Conclusiones generales	56
Recomendaciones.....	56
Referencias bibliográficas	58
Bibliografía	60
Anexos	64

Introducción

Los avances tecnológicos son, desde su surgimiento, el fruto de la inteligencia del hombre, que con el transcurso del tiempo ha sido capaz de analizar y comprender los fenómenos de la naturaleza a su alrededor. La informática es uno de esos principales logros obtenidos, que en la actualidad ocupa un lugar imprescindible en nuestras vidas, pues es casi imposible imaginarnos algún fenómeno o proceso que no esté modelado o soportado por alguna aplicación informática.

Hoy día, los ordenadores son comunes en muchos de los hogares, y sus velocidades de funcionamiento y capacidad hacen parecer maquetas de la edad de piedra aquellos primeros y antiguos ordenadores, permitiéndonos almacenar grandes cantidades de datos e informaciones o simular el comportamiento de un proceso o sistema, y han iniciado una nueva era de la comunicación entre los hombres, sin importar las distancias. Estos indiscutibles avances hacen que muchas personas puedan trabajar desde casa haciendo uso de las facilidades de Internet.

Acercándonos a la digitalización de procesos, se puede afirmar que en los últimos tiempos este fenómeno ha tomado un especial auge en ámbitos diversos como el empresarial, el económico, el ocio, la salud, etc., todo esto condicionado por la movilidad y la flexibilidad que brindan las nuevas tecnologías, de modo que es posible tener acceso a toda la información desde cualquier lugar de forma segura, rápida y precisa.

En Cuba la informatización en empresas, instituciones de salud y educativas ha venido desarrollándose desde hace varias décadas, sumándose así al grupo que usan la informática para beneficio de sus procesos, no solo por no quedarse rezagadas en materia de desarrollo tecnológico, sino por las bondades económico-sociales que la tecnología provee.

Una de las principales tareas desde el surgimiento de la conciencia económica en el hombre fue el control de los recursos de valor que poseía; a medida que iba incrementándose la complejidad de la sociedad fue necesario buscar formas más eficientes para controlar estos recursos.

Actualmente, los procesos económicos en países, empresas y universidades son llevados y controlados de formas diversas, ya sea de forma manual o

digital. La Universidad de Holguín es una institución docente perteneciente al Ministerio de Educación Superior (MES), la cual cuenta con una Dirección Económica que controla la economía, contabilidad y el presupuesto de la entidad.

El control de la economía está dado por la asignación de divisas a la universidad por el ministerio, el cual es el resultado de la aprobación del Plan de la Economía, que declara la necesidad de divisas para su funcionamiento.

Anualmente la universidad recibe un financiamiento en CUC (60% de lo captado por diversas fuentes de ingreso como pregrado compensado, postgrado internacional y eventos), que de acuerdo al Plan de la Economía aprobado por el MES, enmarca los destinos de los gastos que en esta moneda se incurre en la institución.

En la actualidad toda la documentación relacionada a este proceso se realiza de forma manual, tanto el control de cheques emitidos como los destinos a que cada uno de ellos corresponde, por lo que existe un gran volumen de información a manejar y resulta engorroso el control y la toma de decisiones asociadas a este, así como la elaboración de informes, análisis y reportes. Al realizar el análisis de este proceso se observa un conjunto de deficiencias que atentan contra su correcto desempeño. Entre estas deficiencias se concentran las siguientes:

- El registro de la información referente al plan económico de la entidad se realiza de forma manual, generando un consumo excesivo de material de oficina.
- Existen demoras en las estadísticas de control del estado actual del Plan de la Economía, provocando al mismo tiempo demoras en la elaboración de los reportes, por lo que las informaciones solicitadas por la Dirección Económica no son satisfechas a tiempo.
- No existe una forma eficiente de almacenar y gestionar los datos históricos del plan de la economía.

Las deficiencias anteriores constituyen la situación Problemática, a partir de la cual se define el siguiente **problema científico**: ¿Cómo favorecer la gestión de información referente al Plan de la Economía de la Universidad de Holguín "Oscar Lucero Moya"?

Este problema sintetiza el **objeto de investigación** en el: proceso de gestión de la información del Plan de la Economía en la Universidad de Holguín "Oscar Lucero Moya".

Para solucionarlo se plantea el **objetivo general** siguiente: desarrollar un sistema informático para la gestión de la información del Plan de la Economía en la Universidad de Holguín "Oscar Lucero Moya".

El objetivo propuesto delimita como **campo de acción**: informatización del proceso de gestión Plan de la Economía en la Universidad de Holguín "Oscar Lucero Moya".

Para guiar la investigación, se formularon las **preguntas científicas** siguientes:

1. ¿Cuáles son los fundamentos teóricos de la gestión de la información que se realiza en el área de Finanzas CUC de la Dirección Económica de la Universidad de Holguín "Oscar Lucero Moya"?
2. ¿Cómo diseñar un sistema informático para la gestión de la información en el área de Finanzas CUC de la Dirección Económica de la Universidad de Holguín "Oscar Lucero Moya"?
3. ¿Será factible y sostenible el sistema informático que se propone?
4. ¿Cómo implementar un sistema informático para para la gestión de la Información en el área de Finanzas CUC de la Dirección Económica de la Universidad de Holguín "Oscar Lucero Moya"?
5. ¿Cumple el sistema los requerimientos para los cuáles fue diseñado?

Para guiar esta investigación y darle respuestas a las preguntas científicas se realizaron las **tareas** siguientes:

1. Elaborar los fundamentos teóricos de la investigación.
2. Diagnosticar la situación existente en el proceso de control del plan de la economía en área de Finanzas CUC da la de la Dirección Económica de la Universidad de Holguín "Oscar Lucero Moya".
3. Estudiar las tendencias y tecnologías actuales para el desarrollo de aplicaciones Web.
4. Realizar la valoración de sostenibilidad del producto informático.
5. Desarrollar el sistema informático.
6. Implantar del sistema informático.

7. Validar la utilidad del producto informático en el entorno de trabajo donde será utilizado.

Para dar cumplimiento a las tareas planteadas se emplearon diferentes métodos investigativos, tanto teóricos como empíricos, los que se muestran a continuación.

Métodos teóricos

El **análisis y síntesis** se utilizó en la elaboración de los fundamentos teóricos, las conclusiones parciales y generales de la investigación, en el procesamiento de la información y en la descomposición de cada uno de los requerimientos del sistema. Además, este método fue de gran utilidad en la valoración de sostenibilidad del producto.

El **enfoque sistémico** fue empleado para descomponer el sistema en partes y establecer así las relaciones entre ellas, lo cual facilitó la organización del trabajo.

La **modelación** posibilitó identificar los procesos y la relación entre ellos.

Métodos empíricos

La **entrevista** ha sido el método empírico más utilizado en la investigación, con el cual se recolectó la información referente al proceso de control de las finanzas en CUC. Dado que el usuario final es el actor más importante en el desarrollo de un sistema informático, se han tenido en cuenta necesidades, criterios y sugerencias para la correcta concepción del sistema.

La **encuesta** fue utilizada para medir la percepción de los usuarios respecto a la calidad del sistema, y de esta forma evaluar el grado de satisfacción obtenido.

La **revisión de documentos** permitió conocer con claridad los datos de interés, para así poderlos procesar de una forma correcta.

Métodos estadísticos

Para determinar el nivel de aceptación del software dentro de la entidad a través del trabajo con por ciento y frecuencia utilizando el Método Delphi para procesar los resultados de encuestas a expertos.

Estructura del documento

Con el objetivo de exponer los resultados de la investigación desarrollada se ha dividido el presente documento en introducción, capítulo 1 y capítulo 2, conclusiones, recomendaciones, bibliografía y anexos.

Capítulo 1. Fundamentos teóricos: está orientado a mostrar los principales conceptos, fundamentos y metodologías utilizadas en el cumplimiento del objetivo trazado.

Capítulo 2. Descripción de la solución: se describe la solución propuesta, se expone la implementación de la metodología de desarrollo de software ICONIX en el desarrollo de la aplicación y se realiza la valoración de sostenibilidad del producto informático.

CAPÍTULO 1: FUNDAMENTOS TEÓRICOS

En este capítulo se ofrece un acercamiento al análisis económico-financiero, una breve descripción de la entidad donde se enmarca la investigación, así como los conceptos fundamentales del proceso de control del plan económico. Por otro lado, también se tratan las definiciones básicas relacionadas con las aplicaciones Web, así como las tecnologías para su desarrollo y el sistema gestor de base de datos utilizado. Se aborda, además, qué metodología de desarrollo de software se seleccionó para guiar el desarrollo de la solución propuesta.

1.1. Breve acercamiento al análisis económico-financiero

Si bien la actividad cambista y de intercambio de capitales existió desde siempre en las sociedades humanas, podemos decir que el siglo XV, con el surgimiento del capitalismo es el momento central para poder hablar de finanzas tal como las conocemos hoy en día. En este momento es en el cual aparecen los bancos, los cambistas, los intermediarios y otros personajes o actores sociales responsables de este tipo de actividades. Al mismo tiempo, es el siglo XX el siglo en el que el capitalismo pasa a centrar su atención casi exclusivamente sobre las actividades financieras, volviéndose estas más importantes que las industriales o mercantiles de otras épocas.

El análisis económico-financiero juega un papel fundamental, ya que estudia el contenido de los indicadores financieros, ofrece la información necesaria para conocer la situación real durante el periodo de tiempo que se seleccione, basándose en los datos estadísticos, contables y financieros de la empresa, le propicia a los directivos lograr controles económicos ágiles con vista a facilitar la toma de decisiones y tener una administración más eficiente, brinda también a la dirección la herramienta necesaria para el control de la gestión empresarial y la toma de decisiones efectivas que permitan rectificar o mejorar el curso de acción sucesivo a corto, mediano y largo plazo. Es también una de las fases de la Función de Control en la dirección de los procesos económicos, la estrecha relación que existe entre ambos, es por que la función de Control registra y analiza el cumplimiento de los planes trazados y acciona en la forma que sea necesario para garantizar su cumplimiento, por tanto estos tres momentos (registro, análisis, y acción) conforman las distintas

fases en que se estructura el proceso de control, existiendo entre ellas una dependencia funcional, en la fase de registro se fijan las normas de funcionamiento y por intermedio de la misma, quedan reflejados los hechos económicos tal y como se desea que se comporten.

1.2. Descripción de la entidad

El proceso de aseguramiento material y financiero en la Universidad de Holguín “Oscar Lucero Moya” es conducido por dos unidades estructurales: la Vicerrectoría de Administración (VRA) y la Vicerrectoría de Economía (VRE).

La VRE está conformada por tres áreas fundamentales: la Dirección Económica (DE), la Dirección de Alojamiento (DA) y la Dirección de Relaciones Internacionales (DRI). Su misión radica en garantizar el aseguramiento financiero de los procesos sustantivos y contribuir al mejoramiento de las condiciones de estudio, trabajo y vida de la comunidad universitaria, que posibilite el desarrollo de los procesos sustantivos con calidad y en el marco de un control interno razonable. También asegura el desarrollo de las Relaciones Internacionales y la comercialización con el fin de promover la imagen de la universidad y el país, así como la búsqueda de recursos materiales y financieros para el aseguramiento de los procesos.

La Dirección Económica de la Universidad de Holguín “Oscar Lucero Moya”, como parte de la Vicerrectoría Económica, tiene sus procesos organizados en cuatro grupos de trabajo, que responden a las características específicas de la entidad como unidad presupuestada. En la Dirección Económica se aglutinan actividades funcionales donde se reciben los datos primarios de las operaciones materiales y financieras que se producen en el centro. La recepción de los datos primarios por parte de las distintas áreas permite clasificar, registrar y presentar la información económica requerida para la toma de decisiones del centro y para la información a los niveles superiores según lo dispone la legislación vigente. Esta área en sentido general es la representación de la economía del centro tanto interna como externa.

La investigación desarrollada centró su atención en el departamento de finanzas (CUC), que es el encargado del control del presupuesto en divisa asignado a la universidad en el Plan de la Economía. En la actualidad toda la información de este proceso se realiza de forma manual, tanto el control de

cheques emitidos como los destinos a que cada uno de ellos corresponde, por lo que hay gran volumen de información a manejar y resulta engorroso el trabajo de control y la toma de decisiones asociadas a este, así como la elaboración de informes, análisis y reportes.

1.3. Conceptos fundamentales

Presupuesto

Herramienta de planificación, coordinación y control de las actividades a ser realizadas por una organización, previsión de gastos e ingresos para un determinado período de tiempo, por lo general un año. Permite a las empresas, los gobiernos, las organizaciones privadas y las familias establecer prioridades y evaluar la consecución de sus objetivos. El presupuesto suele utilizarse como herramienta para la toma de decisiones sobre la gestión y el crecimiento de la actividad empresarial.

Finanzas

Actividades relacionadas con los flujos de capital y dinero entre individuos, empresas o Estados. Por extensión también se denomina finanzas al estudio de esas actividades como especialidad de la Economía y la Administración, que estudia la obtención y gestión, por parte de una compañía, individuo o del Estado, de los fondos que necesita para cumplir sus objetivos y de los criterios con que dispone de sus activos. En otras palabras, estudia lo relativo a la obtención y gestión del dinero y de otros valores. Las finanzas tratan, por lo tanto, de las condiciones y oportunidad en que se consigue el capital, de los usos de éste y de los pagos e intereses que se cargan a las transacciones en dinero.

Finanzas CUC

Área perteneciente a la Dirección Económica de la Universidad de Holguín cuya función radica en el manejo, gestión y control del Plan de la Economía.

Plan de la Economía

Asignación de divisas a la Universidad de Holguín “Oscar Lucero Moya” por el MES

Cheque

Documento de valor en el que la persona que es autorizada para extraer dinero de una cuenta (por ejemplo, el titular), extiende a otra persona una autorización para retirar una determinada cantidad de dinero de su cuenta, prescindiendo de la presencia del titular de la cuenta bancaria.

1.4 Sistemas Gestores de Información

A continuación se introducen los conceptos relacionados a la gestión de información.

Gestión de información (GI)

La gestión es el conjunto de trámites que se llevan a cabo para resolver un asunto o concretar un proyecto. La gestión de información es el proceso que se encarga de suministrar los recursos necesarios para la toma de decisiones, así como para mejorar los productos, servicios y negocios de una organización [4]. Esta constituye el conjunto de tareas ejecutadas para controlar, almacenar y posteriormente utilizar apropiadamente la información procesada por cualquier institución en el desarrollo de sus actividades. Además, permite trabajar en función de erradicar las dificultades y en aras de perfeccionar el desempeño laboral.

Software de gestión

Se definen como software de gestión aquellas aplicaciones diseñadas para sustituir procedimientos que pueden ser comerciales o administrativos, y que son normalmente realizados por una persona de modo presencial en una empresa o institución, mediante un sistema que posibilita la realización de varios procedimientos a un usuario de forma no presencial, y disminuyendo el esfuerzo de estos.

Funciones de la gestión

Las funciones de la gestión son aplicables para cualquier sistema que se esté desarrollando. Las mismas son un tanto complejas y están estrechamente relacionadas entre sí.

- **Planificar:** determinar qué se va a hacer, decisión que incluye el esclarecimiento de objetivos, establecimiento de políticas, fijación de programas y campañas, determinación de métodos y procedimientos específicos.
- **Organizar:** agrupar las actividades necesarias para desarrollar los planes.

- **Dirigir:** presentar instrucciones y asignar los programas a los responsables de llevarlos a cabo.
- **Controlar:** vigilar si los resultados prácticos se conforman lo más exactamente posible a los programas. Implica estándares, conocer la motivación del personal a alcanzar estos estándares, comparar los resultados actuales con los anteriores y poner en práctica la acción correctiva cuando la realidad se desvía de la previsión [5].

Cada una de estas funciones desempeña un papel determinado dentro del proceso de gestión. Entre ellas existe una estrecha relación, y en caso de que se ajusten estas observaciones al sistema, se concreta el modo de lograr eficiencia y eficacia en dicho proceso.

1.5. Software de gestión de información en el control económico-financiero

En el mundo son muchos los sistemas informáticos que se han desarrollado para gestionar y procesar la información referente al control de finanzas y presupuestos. Se analizaron varios de estos sistemas existentes, en busca de características similares al software que se pretende implementar, y como resultado se obtuvieron los que se mencionan a continuación.

La Facultad de Ciencias Económicas de la Universidad de Holguín cuenta con una aplicación informática diseñada en Excel que realiza el análisis económico-financiero. Dicho sistema fue diseñado solo para ser explotado por sociedades cooperativas y de producción agrícola. No automatiza el punto de equilibrio.

El Banco de Crédito y Comercio (BANDEC) cuenta con un sistema informático que realiza el análisis económico-financiero para su uso interno, es decir, su objetivo es valorar la situación de una entidad cuando ésta solicita un préstamo bancario y de esta forma conocer la capacidad de pago de dicha empresa.

En la red de redes fueron localizados dos sistemas que informatizan los análisis económico-financieros:

“Managerial Analyzer es una herramienta creada para el análisis empresarial, que permite la confección, interpretación y realización del estudio y análisis total de la situación económico-financiera de una empresa o negocio” [6]. Es un sistema muy completo en lo que a análisis se trata, pero está basado en la utilización de Access como gestor de bases de datos, lo que lo lleva a ser un

software con pobres garantías de seguridad de los datos, mucha lentitud y escasez de las prestaciones que brindan los SGBD basados en la arquitectura cliente-servidor. Se basa en los Planes Generales de la Contabilidad española, lo que hace imposible su utilización en nuestro país. Para su adquisición debe ser comprado.

En España se creó un software de análisis de estados financieros empresariales llamado Fin Analysis [7]. Este sistema adolece de muchísimas de las prestaciones deseadas por cualquier empresario. Está basado en Excel, por lo que es imposible totalmente el trabajo en red y la distribución rápida de la información. Los datos deben ser introducidos solo de forma manual, lo que hace engorrosa esta tarea. No existe seguridad de los datos introducidos. Adolece de varios de los métodos empleados en nuestro país para el análisis económico-financiero. Al igual que Manageryal Analyzer, este software también se basa en el plan contable español. En la búsqueda realizada no fue posible constatar la existencia de un sistema que permita la automatización de los análisis económico-financieros dirigidos a los directivos.

Sistemas contables-financieros certificados

En la actualidad existen 38 [8] sistemas certificados para la gestión económica y financiera en Cuba entre los que se encuentran Assets Premium, Siscont 5, Versat Sarasola, Rodas, Centai, entre otros.

El Siscont 5, de la empresa Tecnomática del Ministerio de la Industria Básica (MINBAS), era para el año 2011 “el más difundido en Cuba: tiene más de 800 aplicaciones. Dicho en otras palabras, hay más de 800 instituciones de variado tipo (...) que basan su gestión en el mismo.” [9]. Sin embargo, hacia mediados de 2012 el Versat Sarasola era “el más extendido, no solo en el país, sino en las misiones médicas presentes en varias naciones donde se presta servicio” [10].

Assets Premium es un sistema de gestión integral estándar y parametrizado que permite el control de los procesos de compras, ventas, producción, taller, inventario, contabilidad, presupuesto, activos fijos, útiles y herramientas y recursos humanos. Como sistema integral todos sus módulos trabajan en estrecha relación, generando, automáticamente, al módulo de contabilidad los

comprobantes de operaciones por cada una de las transacciones efectuadas, esto permite que se pueda trabajar bajo el principio de contabilidad al día.

Dispone, además, de métodos novedosos para administración y planificación de inventarios, así como una amplia gama de análisis y consultas que permiten conocer exactamente la situación actual y proyectar decisiones futuras.

Es un sistema flexible, amigable, con ayuda en línea que puede ser instalado en una microcomputadora o sobre varias, funcionando en ambiente multiusuario incluidas estaciones remotas. Asimismo, proporciona opciones de seguridad que le permiten limitar el acceso a los diferentes procesos del sistema de acuerdo con el perfil de cada usuario.

Assets Premium facilita el uso de la parametrización para adaptarse a las exigencias de cada entidad en particular, garantizando que sus reportes tengan la forma y el contenido que el usuario les defina.

Está diseñado con una estructura organizativa a varios niveles, en la que podrán existir: Grupo Corporativo, Corporativo, Grupo de Agrupaciones, Agrupación, Almacenes y Centros de Costos. Para entidades con esta estructura se brinda un Módulo de Comunicaciones que facilita poder intercambiar información entre ellas, con el fin de consolidar información sobre la gestión comercial y contable, pudiéndose obtener los estados financieros, resúmenes de compras, ventas, etc. a distintos niveles.

Actualmente el Assets Premium es el sistema que utiliza la Universidad de Holguín para la gestión de la información contable y de recursos humanos, brindamucas funcionalidades pero no precisamente la que se desea implementar. Estos sistemas son los encargados de generar los estados financieros mencionados anteriormente los que sirven de base para realizar el análisis económico-financiero.

1.6. Tendencias y tecnologías actuales

En la actualidad existen diversas tecnologías que pueden ser utilizados para el desarrollo e implementación de un software. Las tecnologías actuales hacen posible que el proceso de desarrollo de un sistema resulte cada vez más fácil. El gran número de herramientas, lenguajes y metodologías existentes hasta el momento ofrece al personal de trabajo la ventaja de poder escoger técnicas y herramientas más poderosas y cómodas en dependencia del objetivo que se

quiera alcanzar.

A continuación se tratan los conceptos fundamentales relacionados con el software libre, la arquitectura de la aplicación, el servidor web, gestor de base de datos, los lenguajes de programación, los framework, el lenguaje de modelado y la metodología de desarrollo de software, especificando las tecnologías y herramientas seleccionadas para la construcción del sistema.

Software Libre

El "software libre" es la denominación del software que respeta la libertad de los usuarios sobre su producto adquirido y, por tanto, una vez obtenido puede ser usado, copiado, estudiado, modificado, y redistribuido libremente. El software libre suele estar disponible gratuitamente, o al precio de costo de la distribución a través de otros medios; sin embargo no es obligatorio que sea así, por lo tanto no hay que asociar software libre a "software gratuito", conservando su carácter de libre, puede ser distribuido comercialmente ("software comercial").

Análogamente, el "software gratis" o "gratuito" incluye en ocasiones el código fuente; no obstante, este tipo de software no es libre en el mismo sentido que el software libre, a menos que se garanticen los derechos de modificación y redistribución de dichas versiones modificadas del programa.

En Cuba se debe migrar por varias razones al software libre, pues ningún país soberano debe estar basado en el uso de una herramienta tecnológica que responda a un monopolio, especialmente en el caso de Cuba, sometida por varias décadas a un férreo bloqueo económico. En cambio el software libre está basado en la igualdad entre los pueblos, y su filosofía es la de compartir el conocimiento. Esta es la principal razón: la independencia.

El factor económico es otra de las razones. Al ser grande el costo de las licencias de los sistemas operativos, sumado al número de computadoras personales que las usan, el gasto sería considerablemente grande, representando un costo que el país no estaría en condiciones de afrontar. Deben tomarse en cuenta también las políticas monopolistas que impiden a usuarios cubanos la descarga de programas y actualizaciones en muchos sitios de Internet [11].

Arquitectura del sistema

La arquitectura del software identifica los elementos más importantes de una aplicación, así como sus relaciones. Esta es necesaria para entender el sistema, organizar su desarrollo y hacerlo evolucionar de forma adecuada.

La arquitectura responde a los requisitos estructurales con los que debe contar un determinado software, sobre todo a los vinculados con cuestiones estéticas. Se recomienda definir la arquitectura del software desde que se comienza a trabajar, ya que se refleja durante todo el ciclo de desarrollo de un sistema y forma parte de la organización y acabado del producto final.

Arquitectura Cliente/Servidor

En el mundo de TCP/IP las comunicaciones entre computadoras se rigen básicamente por lo que se llama modelo Cliente-Servidor, éste es un modelo que intenta proveer usabilidad, flexibilidad y escalabilidad en las comunicaciones. Desde el punto de vista funcional, se puede definir la computación Cliente/Servidor como una arquitectura distribuida que permite a los usuarios finales obtener acceso a la información en forma transparente aún en entornos multiplataforma.

Entre las principales características de la arquitectura cliente servidor se encuentran:

- El servidor presenta a todos sus clientes una interfaz única y bien definida.
- Es multiplataforma. La plataforma del sistema operativo del cliente y del servidor no son siempre la misma.
- El acceso a los datos es centralizado, lo que evita tener información duplicada o no sincronizada en todos los clientes, además se puede obtener de forma inmediata.
- Permite el uso tanto de hardware como de software de diversos fabricantes, lo cual contribuye considerablemente a la reducción de costos y favorece la flexibilidad en la implantación y actualización de soluciones.

- Las reglas de la organización y las reglas de seguridad se pueden definir una sola vez en el servidor para todos los usuarios.
- Escalabilidad: se puede aumentar la capacidad de clientes y servidores por separado.

Dentro del desarrollo de aplicaciones informáticas una de las más utilizadas es la arquitectura Cliente/Servidor, no solo por las características anteriormente mencionadas, también porque permite la utilización de los servidores con los que cuenta la entidad. La arquitectura reduce los costos de hardware, además proporciona la posibilidad de conectar clientes y servidores independientemente de sus plataformas todo esto conllevó a la aplicación de este modelo en el producto informático que se propone.

Patrón de arquitectura de software

Un patrón de arquitectura de software describe un problema particular y recurrente del diseño, que surge en un contexto específico y presenta un esquema genérico y probado de su solución [12].

Modelo Vista Controlador (MVC) como patrón de arquitectura de software

Modelo Vista Controlador es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. Frecuentemente se ve en aplicaciones web, donde la vista es la página HTML y el código es el que provee de datos dinámicos a la página, el modelo es el Sistema de Gestión de Base de Datos y el controlador representa la lógica de negocio.

- Modelo: es la representación específica de la información con la cual el sistema opera. La lógica de datos asegura la integridad de estos y permite derivar nuevos datos. Encapsula los datos y las funcionalidades.
- Vista: es usualmente la interfaz de usuario, es decir, muestra la información a este y presenta el modelo en un formato adecuado para interactuar.
- Controlador: Responde a eventos que codifican los movimientos o pulsaciones de botones del ratón, teclas, etc., usualmente acciones del usuario, e invoca posibles cambios en el modelo y en la vista [13].

Tanto la vista como el controlador dependen del modelo, el cual no depende de las otras clases. Esta separación permite construir y probar el modelo,

independientemente de la representación visual. El patrón de arquitectura de software MVC tiene flexibilidad para cambiar las vistas y los controladores. Además, es muy adaptativo.

Servidor web

Un servidor web o servidor de protocolo de transferencia de hipertexto HTTP, por sus siglas en inglés es un programa informático que procesa una aplicación del lado del servidor, realizando conexiones bidireccionales y/o unidireccionales con el cliente y generando una respuesta. El código recibido por el cliente suele ser interpretado por un navegador web. Para la transmisión de todos estos datos suele utilizarse algún protocolo, que generalmente es HTTP. El término también se emplea para referirse al ordenador que ejecuta el programa.

Servidor Web Apache

Servidor Web Apache es un servidor HTTP de código abierto, sólido y para uso comercial. Fue desarrollado dentro del proyecto HTTP Server (http) por Apache Software Foundation. Su objetivo principal es proporcionar un servidor eficaz, seguro y extensible con servicios HTTP en conformidad con los estándares actuales de HTTP. Se ejecuta en millones de servidores de Internet y ha sido probado exhaustivamente por desarrolladores y usuarios. **Características del**

servidor web

- Es un servidor web potente, flexible y compatible con HTTP/1.1.
- Implementa los protocolos más recientes, incluyendo HTTP/1.1 (RFC2616).
- Es configurable y extensible con módulos de terceros.
- Es multiplataforma.
- Puede personalizarse utilizando el módulo Apache API.
- Proporciona código fuente completo y viene con una licencia ilimitada.
- Se desarrolla activamente.

Se decidió utilizar el Apache debido a que es libre, de código abierto, y se adapta mejor a las características del hardware existente.

Sistema gestor de base de datos

Un sistema gestor de base de datos (SGBD) se define como el conjunto de programas que administran y gestionan la información contenida en una

base de datos. Ayuda a realizar las siguientes acciones sobre los datos: control de la seguridad y privacidad, definición, mantenimiento de la integridad, y manipulación [14].

El sistema gestor de base de datos es una aplicación que permite a los usuarios definir, crear y manipular una base de datos, además de proporcionar acceso controlado a la misma de forma eficiente.

MySQL

Este sistema fue creado por la empresa sueca MySQLAB, con licencia GPL. Su diseño multihilo le permite soportar una gran carga de forma muy eficiente. Es muy utilizado en aplicaciones Web, porque existe baja concurrencia en la modificación de datos y en cambio el entorno es intensivo en lectura de datos. Es muy rápido, confiable, robusto y fácil de usar. Además tiene un conjunto muy práctico de características desarrolladas en estrecha cooperación con los usuarios. A través de un constante desarrollo, MySQL ha logrado ofrecer un variado conjunto de funcionalidades. La conectividad, velocidad y seguridad que brinda hace que este gestor sea altamente conveniente para acceder a bases de datos en Internet.

Algunas de sus principales características son:

- Posee gran portabilidad entre sistemas.
- Soporta gran cantidad de tipos de datos para las columnas.
- Aprovecha la potencia de sistemas multiprocesador, gracias a su implementación multihilo.
- Dispone de API's en gran cantidad de lenguajes (C, C++, Java, PHP, Python).
- Su bajo consumo lo hace apto para ser ejecutado en una máquina con escasos recursos.
- Ofrece facilidades de instalación y configuración.

Se tomó la decisión de utilizar MySQL en su versión 5.5 la cual a diferencia de sus sucesivas versiones es libre. Además se consideró la recomendación del usuario de que este fuese el SGBD utilizado.

Lenguajes de programación

Un lenguaje de programación es un idioma artificial diseñado para describir el conjunto de acciones consecutivas que un equipo debe ejecutar. Este está formado por un grupo de símbolos, sintaxis y reglas semánticas que definen su estructura, así como el significado de sus elementos y expresiones. Los lenguajes de programación disponen de formas adecuadas para implementar, que permiten ser leídas y escritas por personas, y a su vez resultan independientes del modelo de computadora a utilizar.

JavaScript

JavaScript es un lenguaje de programación utilizado principalmente para crear páginas web dinámicas. El navegador del cliente es el encargado de interpretar las instrucciones Javascript y ejecutarlas para realizar efectos especiales e interactividades con el usuario, de modo que el mayor recurso, y tal vez el único, con que cuenta este lenguaje es el propio navegador.

El desarrollo en Javascript es el siguiente paso, después del HTML, que puede dar un programador que decida mejorar sus páginas y aumentar la potencia de sus proyectos. Este lenguaje puede ser aprendido con facilidad y utilizado con solo un poco de práctica, ya que es un lenguaje de programación bastante sencillo.

Entre las acciones típicas que se pueden realizar en Javascript existen dos vertientes. Por un lado los efectos especiales sobre páginas web, para crear contenidos dinámicos y conseguir que los elementos de la página que tengan movimiento, cambien de color, o cualquier otro dinamismo. Por el otro, Javascript permite ejecutar instrucciones como respuesta a las acciones del usuario, con lo que se pueden crear páginas interactivas con programas como calculadoras, agendas, o tablas de cálculo.

Javascript es un lenguaje con muchas posibilidades, permite la programación de pequeños *scripts*, pero también de programas más grandes, orientados a objetos, con funciones, estructuras de datos complejas, etc. Además, Javascript pone a disposición del programador todos los elementos que forman la página web, para que este pueda acceder a ellos y modificarlos dinámicamente [15].

Técnicamente, Javascript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. Los programas escritos con este lenguaje se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios. A pesar de su nombre, Javascript no guarda ninguna relación directa con el lenguaje de programación Java. Legalmente, Javascript es una marca registrada de la empresa Sun Microsystems.

Python

Python es un lenguaje de alto nivel cuya filosofía hace hincapié en una sintaxis muy limpia y que favorezca un código legible. Se trata de un lenguaje de programación multiparadigma, que facilita el hecho de que los programadores no tengan que ser forzados a un estilo en particular, ya que soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional. Otros paradigmas están soportados mediante el uso de extensiones. Es un lenguaje interpretado, fuertemente tipado, usa tipado dinámico, implementa el conteo de referencias para la administración de memorias, y es multiplataforma [16].

Es un lenguaje de programación de propósito general, cuya expansión y popularidad es relativamente reciente. Este lenguaje es una apuesta por la simplicidad, versatilidad y rapidez de desarrollo. Preparado para realizar cualquier tipo de programa, desde aplicaciones Windows a servidores de red o incluso, páginas web.

Una característica importante de Python es la resolución dinámica de nombres; es decir, lo que enlaza un método y un nombre de variable durante la ejecución del programa (también llamado ligadura dinámica de métodos).

Otro objetivo del diseño del lenguaje es la facilidad de extensión. Se pueden escribir nuevos módulos fácilmente en C o C++. Python puede incluirse en aplicaciones que necesitan una interfaz programable [16].

Python es una interesante opción para realizar todo tipo de programas que se ejecuten en cualquier máquina. El equipo de desarrollo está trabajando de manera cada vez más organizada y cuentan con el apoyo de una comunidad que está creciendo rápidamente. Algunas empresas que utilizan Python son Yahoo, Google, Walt Disney, NASA, Red Hat, etc.

En los últimos años el lenguaje se ha hecho muy popular, gracias a varias razones como:

- La cantidad de librerías que contiene, tipos de datos y funciones incorporadas en el propio lenguaje, que ayudan a realizar muchas tareas habituales sin necesidad de tener que programarlas desde cero.
- Un programa en Python puede tener de 3 a 5 líneas de código menos que su equivalente en Java o C.
- La cantidad de plataformas en las que es posible desarrollar, como Unix, Windows, OS/2, Mac, Amiga y otros.

¿Por qué utilizar Python como lenguaje de programación?

Las características fundamentales que conllevan a seleccionar como lenguaje de programación a Python se listan a continuación:

- Se pueden crear todo tipo de programas. No es un lenguaje creado específicamente para la web, aunque entre sus posibilidades se encuentra el desarrollo de páginas.
- Python dispone de un intérprete por línea de comandos en el que se pueden introducir sentencias. Cada sentencia se ejecuta y produce un resultado visible, lo cual resulta de gran utilidad para entender mejor el lenguaje y probar los resultados de porciones de código rápidamente.
- Python es muy legible, la sintaxis de Python es muy elegante y permite la escritura de programas cuya lectura resulta fácil, en comparación con otros lenguajes.

Ventajas

- Desarrollo más rápido: Es posible escribir un programa, salvarlo y ejecutarlo, sin necesidad de pasar por un proceso lento y extensivo de compilación.
- Multiplataforma: el mismo código funciona en cualquier arquitectura, la única condición es que se disponga del intérprete del lenguaje. No es necesario compilar el código una vez para cada arquitectura.
- Cuenta con intérpretes gratuitos (ejemplo Eclipse)
- Python es gratuito, incluso para propósitos empresariales.
- Python es fácil de aprender, provee una mezcla única de elegancia, simplicidad, sentido práctico, y poder.

Desventajas

- **Lentitud:** Los programas interpretados son más lentos que los compilados. Sin embargo los programas interpretados suelen ser cortos, en los que la diferencia es despreciable.

Python es un lenguaje de programación de uso general. Este lenguaje estable. Ofrece alto nivel de productividad para todas las fases del ciclo vital del software: Análisis, diseño, prototipado, codificación, experimentación, corrección de errores, documentación, implementación, y, por supuesto, mantenimiento. La popularidad de Python ha tenido un crecimiento sostenido a través de los años. En la actualidad, familiarizarse con Python es una ventaja para cada programador, sirviendo como parte importante en cualquier solución de software.

Framework Web

Un Framework simplifica el desarrollo de una aplicación mediante la automatización de algunos de los patrones utilizados para resolver las tareas comunes. Este proporciona estructura al código fuente, forzando al desarrollador a crear código más legible y más fácil de mantener. Además, este facilita la programación de aplicaciones, ya que encapsula operaciones complejas en instrucciones sencillas [17].

Los frameworks ayudan en el desarrollo de software, proporcionan una estructura definida, la cual ayuda a crear aplicaciones con mayor rapidez. Ayuda a la hora de realizar el mantenimiento del sitio gracias a la organización durante el desarrollo de la aplicación. Los frameworks son desarrollados con el objetivo de brindar a los programadores y diseñadores una mejor organización y estructura a sus proyectos.

Django

Django es un framework de desarrollo web de código abierto, escrito en Python. Sigue el patrón MVC (Modelo-Vista-Controlador).

Es un framework de alto nivel y de código abierto, escrito en el lenguaje de programación Python, con el objetivo de garantizar un desarrollo web rápido y limpio, con un diseño pragmático. La meta fundamental de Django es facilitar la creación de sitios web complejos. Django pone énfasis en la reutilización, la conectividad y extensibilidad de componentes. Cada una de las partes del

framework Django ha sido diseñada con el concepto de productividad en mente [19].

Otras características de Django son:

- Un mapeador objeto-relacional (ORM por sus siglas en inglés).
- Aplicaciones que pueden instalarse en cualquier página gestionada con Django.
- Una API de base de datos robusta.
- Un sistema incorporado de "vistas genéricas" que ahorra tener que escribir la lógica de ciertas tareas comunes.
- Un sistema extensible de plantillas basado en etiquetas, con soporte de herencia.
- Un sistema de "middleware" para desarrollar características adicionales; por ejemplo, la distribución principal de Django incluye componentes middleware que proporcionan cacheo, compresión de la salida, normalización de URLs, protección falsificación de petición en sitios cruzados (CSRF por sus siglas en inglés) y soporte de sesiones.
- Soporte de internacionalización, incluyendo traducciones incorporadas.
- Interfaz de administración.
- Documentación incorporada accesible a través de la aplicación administrativa (incluyendo documentación generada automáticamente de los modelos y las bibliotecas de plantillas añadidas por las aplicaciones).

Se decidió utilizar Django debido al corto tiempo en el ciclo de desarrollo de aplicaciones, y por las características antes mencionadas.

Metodologías de desarrollo de software

Desde el punto de vista informático una metodología de desarrollo de software es el conjunto de procedimientos, técnicas, herramientas y un soporte documental a la hora de desarrollar un producto de software, que indica quién, cuándo y cómo hacer algo [20].

Actualmente existen muchas metodologías de desarrollo de software que pueden ser aplicadas en dependencia del sistema que se desea desarrollar. Entre las más destacadas a nivel mundial se encuentran XP (eXtreme

programming, por sus siglas en inglés), RUP (Rational Unified Process, por sus siglas en inglés) e Iconix.

Metodología Iconix

El desarrollo ágil ha detonado el mundo del desarrollo de software, propiciando el cambio en la forma en que se desarrollan los productos informáticos. Tiene mucho sentido porque está dirigido hacia las razones por las cuales los proyectos han fallado durante años.

Durante 1995, el Standish Group's CHAOS Report (muy citado por enfocarse en las razones de fallas y éxitos de los proyectos) mostró que la primera razón por la que un proyecto no tiene éxito es por la carencia de las opiniones de los usuarios. Este estudio identificó los elementos esenciales que cambian positivamente cuando se involucra a los usuarios, por lo que no es sorpresa que se involucre al usuario, y en particular, la retroalimentación temprana del usuario, como conductor sumamente importante detrás de la agilidad.

Otro elemento muy considerado entre las fallas en los proyectos es “el síndrome de los grandes proyectos monolíticos“, donde el proyecto no alcanza ver la luz por un año o dos. Finalmente cuando esta monstruosidad se desprende de los programadores, la decepción de los clientes aumenta y no se conoce que sucedió con las especificaciones originales, no es aceptado por los clientes que dieron sus consideraciones, y el producto es completamente diferente debido a la interpretación de los programadores, o porque simplemente se cambiaron los requisitos pasados los dos años y se olvidó comunicarlo a los programadores.

Iconix process se encuentra en un espacio entre el muy amplio Proceso Unificado de Rational (RUP) y el muy pequeño enfoque de programación extrema (XP). Iconix Process es dirigido por casos de uso como RUP, pero sin la excesiva cantidad que RUP propone. Es relativamente pequeño y ligero, como XP, pero no desprecia el análisis y el diseño como XP hace. Este proceso realiza un uso estilizado de Lenguaje Unificado de Modelado (UML por sus siglas en inglés), manteniéndose enfocado en la trazas de los requisitos. El proceso está en consonancia con la visión original de Ivar Jacobson de lo que significa “dirigido por casos de uso”.

Este enfoque toma lo mejor de las tres metodologías que han existido desde principio de los 90's. Estas metodologías fueron desarrolladas por Ivar Jacobson, Jim Rumbaugh, y Grady Booch. Utiliza un subconjunto de elementos de UML basados en el análisis de Doug de estas tres metodologías

Si se analizan comparativamente modelos de proceso como RUP, XP y SCRUM con respecto a Iconix, y se tienen en cuenta las características del proyecto, como el tamaño del proceso, del equipo y la complejidad del problema, para cada uno de los modelos, se puede resaltar que con un pequeño equipo de desarrollo se puede realizar proyectos de tamaño pequeño o medio, con una complejidad pequeña o media.

Con respecto a la curva de aprendizaje, vemos que los modelos ágiles ofrecen una mayor ventaja, aunque con ciertas limitaciones, ya que aún no han sido explotados a gran escala como lo es RUP, que posee alto soporte y herramientas integrales que guían a los desarrolladores a través de este, facilitando aplicar con mayor efectividad esta metodología, y permitiendo aprovecharla al máximo. Sin embargo Iconix posee algún soporte disponible tanto de herramientas de integración como de soporte externo.

Tareas de Iconix Process

A continuación se mencionan las tareas que se llevan a cabo como parte de la metodología Iconix Process para la construcción de un sistema informático, en aras de crear una visión global del proceso.

1. Análisis de requisitos: Implica asegurar que los casos de uso y el modelo de dominio trabajen juntos y dirigidos a los requisitos funcionales de los clientes. También implica que los clientes tengan una idea clara de que el equipo de desarrollo tiene la capacidad de realizar un diseño sobre la base de lo que el cliente quiere y satisfacer sus requerimientos. Algunas escuelas plantean que "los clientes nunca saben lo que quieren... Los requisitos cambian semanalmente, a veces a diario, incluso en horas," y se ha utilizado esto como escudo para justificar omisiones en el análisis o el diseño. Este es un gran error. Es el trabajo del analista ayudar a los clientes en el entendimiento de sus requerimientos, los casos de uso, prototipos, modelos de dominio, entre otras, las herramientas que se utilizan para ello.

2. Análisis y diseño preliminar: requiere de la realización de los diagramas de robustez y la documentación de los casos de uso para cada escenario que se planea construir, y asegurar que los diagramas y la descripción textual de los casos de uso se correspondan unos con otros, que ambos estén completos y representen correctamente el comportamiento deseado para el sistema. Esto significa también asegurar que el modelo del dominio se corresponda con los diagramas de robustez en particular, y que todos los objetos entidad que se muestran en los diagramas de robustez estén representadas en el modelo de dominio. En otras palabras, es necesario verificar que se han identificado las abstracciones principales del espacio del problema que se necesitan para implementar el comportamiento deseado.

También se debe revisar si estas clases entidades tienen atributos y si se puede seguir el flujo de datos entre las pantallas del sistema; ya sea a través de las clases o quizás en las tablas de la base de datos donde se almacena la información persistente. Se debe revisar la arquitectura técnica detrás del diseño evolutivo y asegurar que el diseño que se ha iniciado sea aceptado en el contexto de esta arquitectura técnica.

3. Diseño detallado: Significa asegurar que el “cómo” del diseño detallado, como muestran los diagramas de secuencia y de clases asociados estén acorde con el “qué” que especifican los casos de uso. Es decir, que el diseño detallado sea lo suficientemente profundo para facilitar un salto relativamente pequeño hacia el código. También significa la revisión de la calidad del sistema desde cierto número de perspectivas. Estas pueden incluir modularidad, cohesión de las clases, acoplamiento entre objetos y otras métricas que se pueden llamar “bondades de la orientación a objetos”.

Al mismo tiempo debe lograr que el diseño se corresponda con el estándar de diseño interno de la organización. A veces este estándar puede lograrse con el uso de patrones de diseño. Se debe realizar el mejor de los intentos de estabilización y validación de los requerimientos y la arquitectura técnica, antes de obtenerlo. El diseño es el último paso antes del código, por lo que en este punto se debe cerrar todo concerniente a estos tópicos.

4. Implementación: En este paso es donde los programadores toman el diseño y codifican a partir de él. Los programadores deben estar involucrados en todas las etapas del diseño (lo ideal es que los diseñadores sean los programadores).

Este paso se subdivide en:

1. Escribir/ generar código.
2. Realizar las pruebas de unidad e integración.

Un elemento importante aquí es que los programadores deben ir realizando sus propias pruebas, incluyendo las pruebas de integración. Es conveniente realizar pruebas de integración por un programador diferente al que codificó el módulo, además de la experiencia del programador original. Esto a menudo ayuda a detectar más errores, y tiene el beneficio añadido de que cada programador se involucra más con el resto de proyecto.

La metodología escogida para guiar el proceso de desarrollo de este trabajo es Iconix Process.

Herramientas CASE

Una buena parte del tiempo de desarrollo de un sistema informático se dedica al diseño. Existen varias herramientas y lenguajes que se han creado para facilitar esa parte del trabajo, como es el caso de la Ingeniería de Software Asistida por Computadoras (CASE por sus siglas en inglés).

CASE no es más que la aplicación científica de un conjunto de herramientas y métodos a un sistema de software, el cual se quiere que resulte en un producto de software de alta calidad, libre de errores y mantenible. También se refiere a los métodos de desarrollo de sistemas de información, junto con herramientas automáticas que pueden usarse en el proceso de desarrollo de software.

El UML es un lenguaje gráfico para visualizar, especificar, construir y documentar artefactos de un sistema de software. Este ofrece una vía estándar para escribir los planos de sistemas informáticos, donde se incluyen aspectos conceptuales como procesos de negocios y funciones de sistemas, o aspectos concretos como sentencias en lenguajes de programación, esquemas de bases de datos y componentes de software reusables.

Cuando se complementa CASE con el lenguaje UML en una aplicación informática se crea lo que usualmente conocemos como herramienta CASE.

Existe una gran variedad de estas en la actualidad, orientada a disímiles usos y metodologías de desarrollo.

Enterprise Architect

Enterprise Architect (EA) es una herramienta comprensible de diseño y análisis UML, cubriendo el desarrollo de software desde el paso de los requerimientos a través de las etapas del análisis, modelos de diseño, pruebas y mantenimiento. Es además una herramienta diseñada para ayudar a construir software robusto y fácil de mantener, ya que ofrece salida de documentación flexible y de alta calidad [22].

Fue escogida por encima de Rational Rose y Visual Paradigm, dos de las herramientas más empleadas, porque a pesar de ser privativa como estas, resulta menos costosa y existen algunas versiones de prueba gratuitas. Enterprise Architect ayuda a visualizar aplicaciones mediante el empleo de la ingeniería inversa desde un amplio rango de lenguajes de desarrollo de software y esquemas de repositorios de base de datos. Además provee una herramienta de estimación de proyectos comprensiva que calcula el esfuerzo desde los objetos de caso de uso, obteniendo resultados tan fiables como los obtenidos a partir del método COCOMO, y en un tiempo considerablemente menor [23].

Conclusiones parciales del capítulo

Con la presentación de los fundamentos teóricos que dan soporte a esta investigación se definieron los conceptos y la forma con que opera el proceso de gestión las finanzas en divisas de la entidad objeto de estudio. Por otra parte se expuso un análisis de algunos software existentes para la gestión económico-financiera y planes económicos culminando el capítulo con los principales elementos tecnológicos relacionados con las aplicaciones Web, el sistema Gestor de bases de Datos y la Metodología de Desarrollo de Software a utilizar.

CAPÍTULO 2. DESCRIPCIÓN DE LA SOLUCIÓN

En este capítulo se hace una descripción de la solución propuesta. Siguiendo los pasos y fases que plantea Iconix, que ha servido para organizar todo el proceso de desarrollo de la aplicación, así como brindar una serie de pasos que permiten concretar la elaboración del producto informático, pues este proceso es una guía que describe como ir desde los casos de uso hasta la codificación del software.

2.1. Modelo del dominio

El modelo del dominio es una parte esencial del proceso de ICONIX. Como su nombre lo indica siempre refleja la comprensión actual del espacio del problema o lo que es lo mismo, define el alcance que tendrá el proyecto. Contiene un glosario de términos utilizados en el proyecto, proporcionando un vocabulario común que posibilita una comunicación clara entre los miembros del equipo de desarrollo. Además forma la base en la cual se construyen los casos de uso [24].

Definición de requerimientos

La definición de requerimientos es un proceso fundamental para el éxito de un proyecto de desarrollo. Comprende la determinación de las necesidades o condiciones que debe satisfacer el producto propuesto como solución.

Requerimientos funcionales

El proceso de desarrollo de ICONIX detallado, plantea que los requerimientos funcionales son una fuente importante a tener en cuenta en el modelado del dominio. El cual se realiza en la construcción del software, ya que contienen la gran mayoría de los términos y conceptos claves utilizados en el contexto donde se involucre la problemática por resolver, como en el caso de la organización para la presente investigación [24].

Por lo que es conveniente registrar, como se muestra a continuación, los principales requerimientos funcionales que debe tener la aplicación, antes de continuar con su elaboración.

Lista de requerimientos funcionales

El sistema debe de ser capaz de:

1. Validar el acceso de los usuarios
2. Gestionar la información de los usuarios del sistema

- 2.1 Insertar usuario
- 2.2 Listar usuario
- 2.3 Modificar usuario
- 2.4 Eliminar usuario
- 2.5 Buscar usuario
- 3. Gestionar Plan de la Economía
 - 3.1 El sistema debe permitir al usuario la creación de un nuevo Plan de la Economía
 - 3.2 El sistema debe permitir al usuario modificar planes de la economía existentes
 - 3.3 El sistema debe mostrar los planes de la economía con los siguientes parámetros (Fecha, Indicador, Total)
 - 3.4 El sistema debe permitir buscar los planes de la economía existentes atendiendo a los siguientes criterios de búsqueda (fecha, Indicador)
 - 3.5 El sistema debe imprimir los planes de la economía con los siguientes parámetros (Fecha, Indicador, Total)
- 4. Gestionar asignaciones.
 - 4.1 El sistema de permitir al usuario la creación de nuevas asignaciones.
 - 4.2 El sistema debe permitir al usuario modificar las asignaciones existentes
 - 4.3 El sistema debe mostrar al usuario las asignaciones existentes de acuerdo a los siguientes parámetros (Fecha, Tipo de Asignación, Total)
 - 4.4 El sistema debe imprimir las asignaciones existentes con a los siguientes parámetros (Fecha, Tipo de Asignación, Total)
 - 4.5 El sistema debe permitir al usuario eliminar las asignaciones existentes
- 5. Gestionar Cheques
 - 5.1 El sistema debe permitir al usuario introducir al sistema los nuevos cheques emitidos
 - 5.2 El sistema debe permitir al usuario modificar los cheques emitidos
 - 5.3 El sistema debe permitir al usuario eliminar los cheques emitidos

- 5.4 El sistema debe ser capaz de mostrar los cheques emitidos con los siguientes parámetros (Fecha, No Cheque, Suministrador, Producto, Clasificación, Importe, Informe de recepción, Estado actual)
 - 5.5 El sistema debe permitir al usuario la búsqueda de cada cheque emitido de acuerdo a los siguientes criterios de búsqueda (Fecha, Suministrador, Clasificación, Estado actual)
6. Gestionar reportes.
- 6.1 Reporte mensual: El sistema debe imprimir los reportes solicitados atendiendo al criterio de búsqueda (mensual)
 - 6.2 Reporte acumulativo hasta la fecha: El sistema debe imprimir los reportes solicitados atendiendo al criterio de búsqueda (acumulativo hasta la fecha)
 - 6.3 Reportes Generales: El sistema debe imprimir los reportes generales de acuerdo a situación actual del cumplimiento del plan de la economía
7. Seguridad
- 7.1 El sistema debe ser capaz de establecer el nivel de acceso de usuarios
 - 7.2 El sistema debe ser capaz de guardar trazas de las operaciones y modificaciones realizadas en la aplicación por los usuarios que interactúan con este.

Requerimientos no funcionales

Después de definir los requerimientos funcionales de la aplicación se hace necesario determinar los no funcionales que van más allá de lo que se quiere conseguir en términos operacionales con la aplicación y que contempla aspectos claves como son usabilidad, seguridad, hardware, portabilidad e interfaz externa. En la medida que estos sean lo más flexible posible ante cambios futuros, intuitivos, agradables a la vista, que favorezcan la acción del experto y no entren en contradicción con la misma y además permitan que la dinámica de trabajo fluya lo más rápido posible, contribuirán o no al nivel de satisfacción de los usuarios.

Lista de requerimientos no funcionales

Apariencia o Interfaz externa

- El sistema debe ser fácil de manipular, pues los usuarios finales no son expertos en computación, a pesar de hacer uso de la computadora para realizar las labores de trabajo.
- El diseño debe ser agradable y atrayente a los usuarios para lograr una mejor concentración, sin desviar demasiado su atención del contenido de trabajo.
- Los colores predominantes son el blanco, el gris y el azul en menor escala. Propiciando una combinación agradable a la vista.
- La interfaz no será cargada con imágenes para proporcionar una navegación cómoda.
- El idioma a utilizar será el español.
- De no poderse ejecutar una acción se visualizará un mensaje de error.

Usabilidad

- El sistema debe ser accesible desde cualquier lugar de la Universidad.
- El diseño del sistema debe ser sencillo para agilizar el tiempo de conexión a este.

Rendimiento

- El tiempo de respuesta en las búsquedas debe ser corto, por lo que el acceso a la Base de Datos (BD) se debe efectuar de forma rápida.

Soporte

- Debe dar facilidad de mantenimiento una vez implantado para posibilitar un perfeccionamiento continuo del sistema.
- Ante cualquier falla el sistema debe dar una respuesta inmediata.

Portabilidad

- Las herramientas utilizadas para el desarrollo del sistema son tecnología de software libre y a su vez multiplataforma, lo cual le confiere al sistema esta última característica.

Seguridad

- Sólo los usuarios autorizados podrán acceder al sistema.
- Debe facilitar el cambio de contraseñas.
- Debe garantizar que las funcionalidades del sistema se realicen de acuerdo al nivel del usuario que esté activo.

- No debe existir información que se pueda obtener sin ser usuario del sistema.
- Sólo el administrador del sistema tendrá acceso a la BD y es responsable de la autorización en general del mismo.
- El sistema debe tener protección contra acciones no autorizadas para evitar afectar la integridad de la información almacenada.

Confiabilidad

- El sistema debe posibilitar la recuperación de información en caso de fallos y/o errores.

Ayuda y documentación en línea

- Debe contar con un Manual de Usuario y un sistema de ayuda de forma tal que le brinde orientación al usuario respecto a las opciones con que cuenta el sistema, utilizando textos explicativos que indiquen la acción de estas.

Software

- Las máquinas de los usuarios deben tener el SO Windows 2000 o superior, deben disponer un sistema operativo con un navegador Web. Se recomienda el Internet Explorer 7.0 o superior, Mozilla FireFox 1.5 o superior u Opera 8 o superior.
- La máquina servidor debe tener instalado Windows Server 2003 o superior, Python 2.7.5, Django 1.5.2, MySQL Server (5.5.8), Apache Server 2.2.17.

Hardware

- Como requerimientos mínimos es necesaria una PC con procesador Pentium 3 o superior. La máquina servidor debe tener 512 RAM o superior y debe ser un Pentium IV con un microprocesador cercano a los 2.6 GHz de velocidad. Espacio en disco duro de 600 MB o más, disponibles para la instalación del software requerido.

Conceptos que integran el Modelo del dominio

Usuario: son todos los actores del sistema.

Concepto: razón que da origen a los gastos en que se incurre en la entidad.

Plan: enmarca los destinos de los gastos que en esta moneda se incurre en la institución.

Elemento de gasto: destino de gasto en que se incurre en la institución.

Asignaciones: Cantidad de dinero asignada por el ministerio a un elemento de gasto.

Partidas: cantidad de dinero real para un posible gasto.

Registros: cheques emitidos de los gastos incurridos por la institución

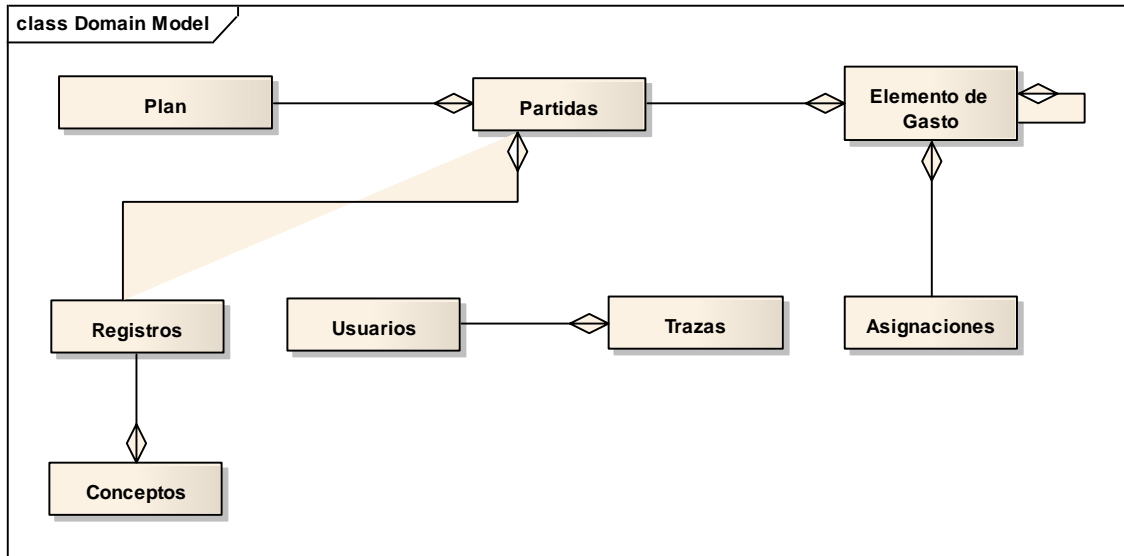


Figura 1. *Diagrama del Modelo del Dominio*

2.2. Modelación de los casos de uso

A partir de los requerimientos funcionales capturados se desarrolla el Modelo de Casos de Uso, que comprende la identificación de los actores y casos de uso del sistema. No obstante, un importante aspecto de ICONIX es que un requisito se distingue explícitamente de un caso de uso; este último describe un comportamiento mientras que un requisito describe una regla para el comportamiento. Los actores representan entidades externas al sistema, y son análogos a un rol del usuario [24].

Actores del sistema

Se puede determinar los actores del sistema examinando a los usuarios que lo utilizarán y a otros sistemas que deben interactuar con él. Cada categoría de usuarios o sistemas que interactúan se representan por tanto como actores [34].

A continuación se muestra el diagrama de actores del sistema.

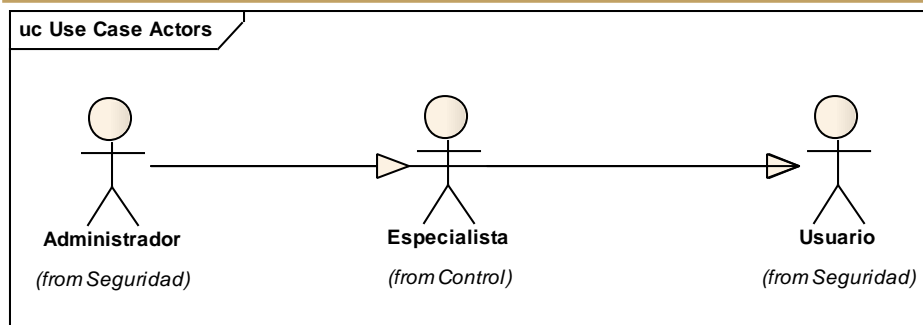


Figura 2. Actores del sistema

Los actores son llamados usuario, aunque le es dado el nombre del rol específico que desempeña en el sistema. A continuación se identificaron los actores siguientes:

Administrador: Es el responsable de la gestión de los usuarios y de la configuración necesaria para el correcto funcionamiento del sistema. Además de controlar las acciones que hace cada usuario y hacer salvadas de la base de datos.

Especialista: persona responsable de controlar el Plan de la Economía aprobado a la Universidad junto al financiamiento CUC asignado a la institución.

ICONIX asume que el Modelo del Dominio inicial es incorrecto y provee un mejoramiento incremental del mismo a medida que se analizan los casos de uso [24]. Para una mejor comprensión del modelo, los casos de uso se agrupan en subsistemas o paquetes, según las relaciones existentes entre ellos, como se muestra en la Figura 3.

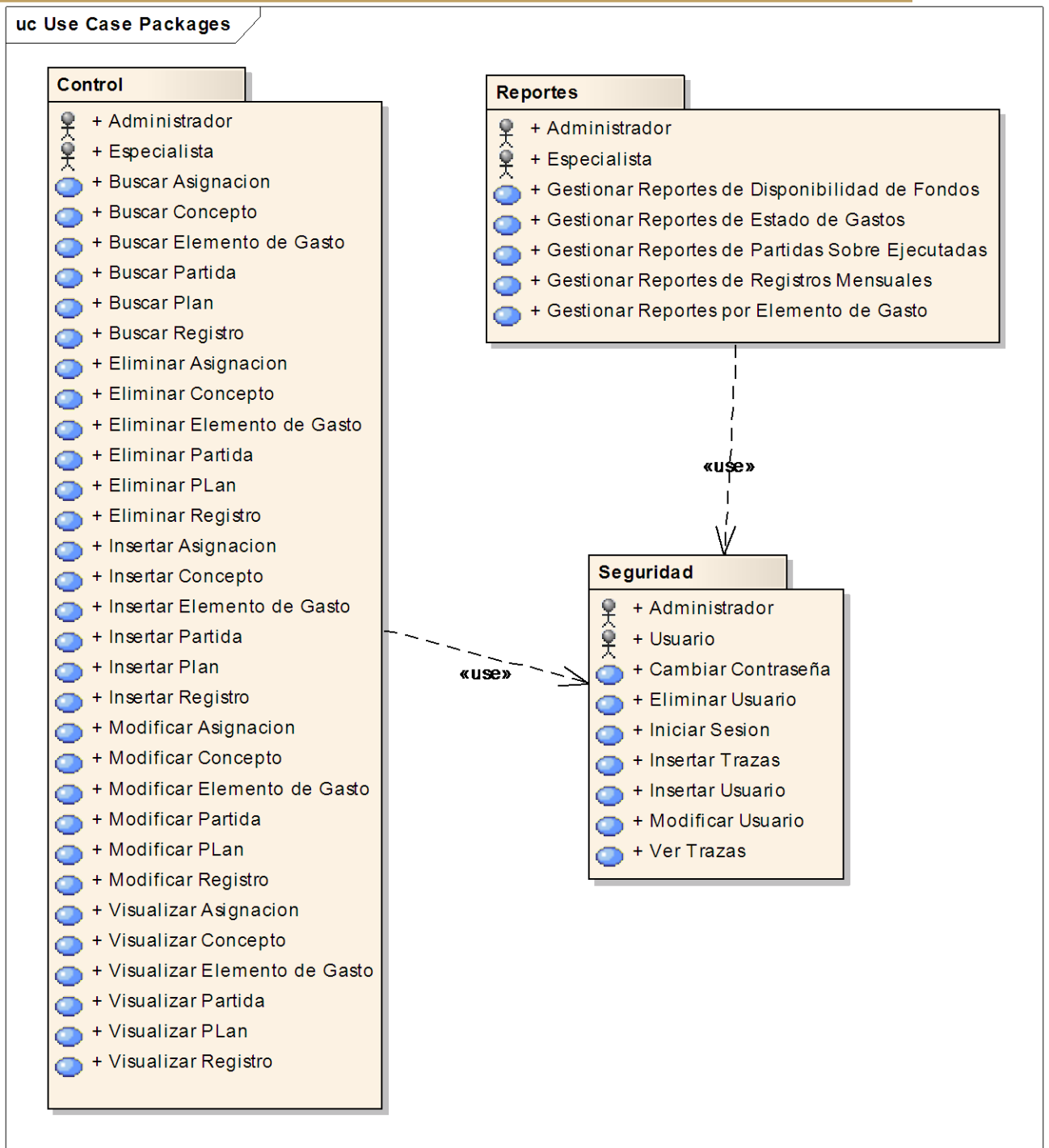


Figura 3. Diagrama de Paquetes

Para cada subsistema o paquete se realiza un diagrama de casos de uso (ver Figura 4), donde se relacionan los actores del sistema con los casos de uso correspondientes.

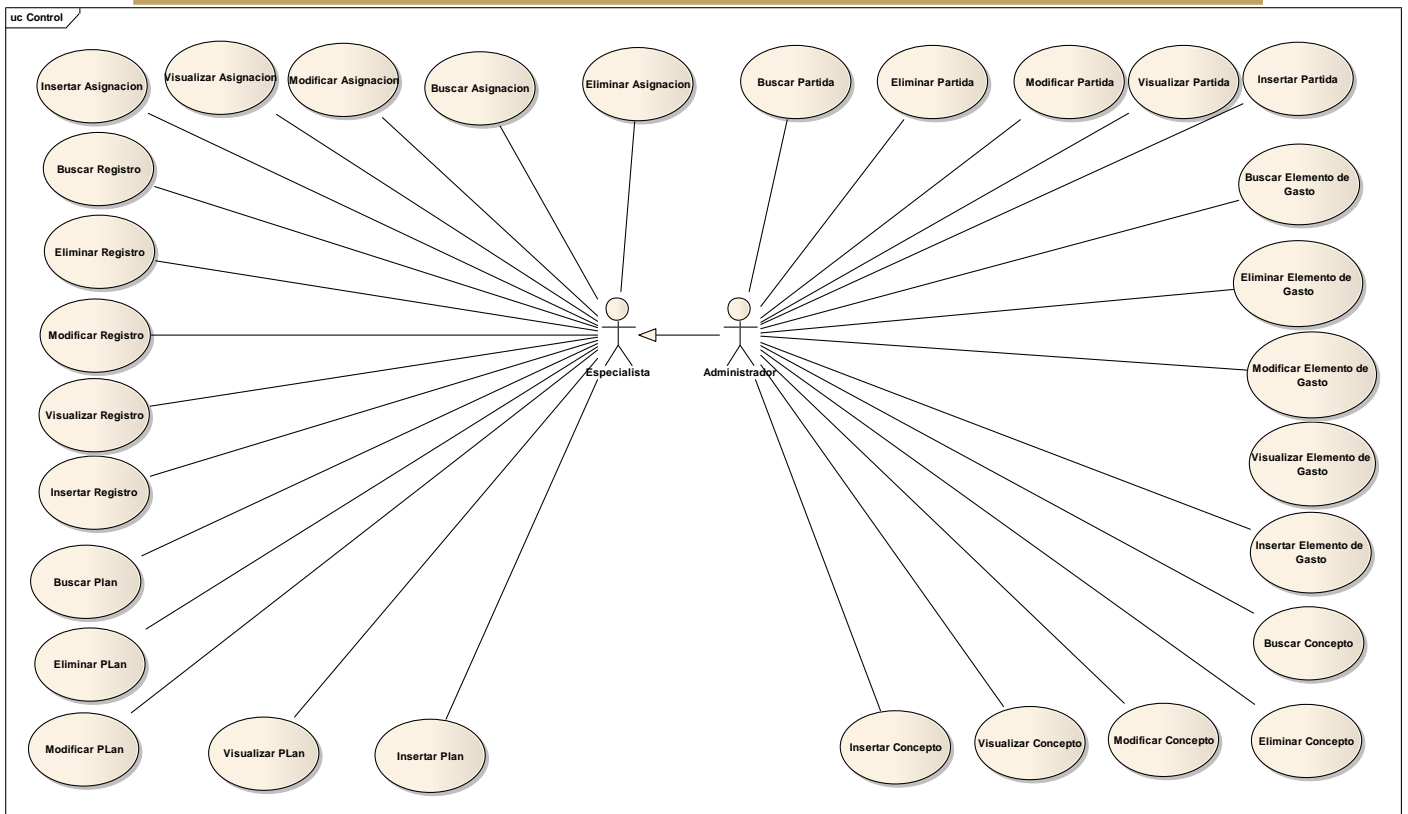


Figura 4. Diagrama de Casos de Usos Paquete de Control

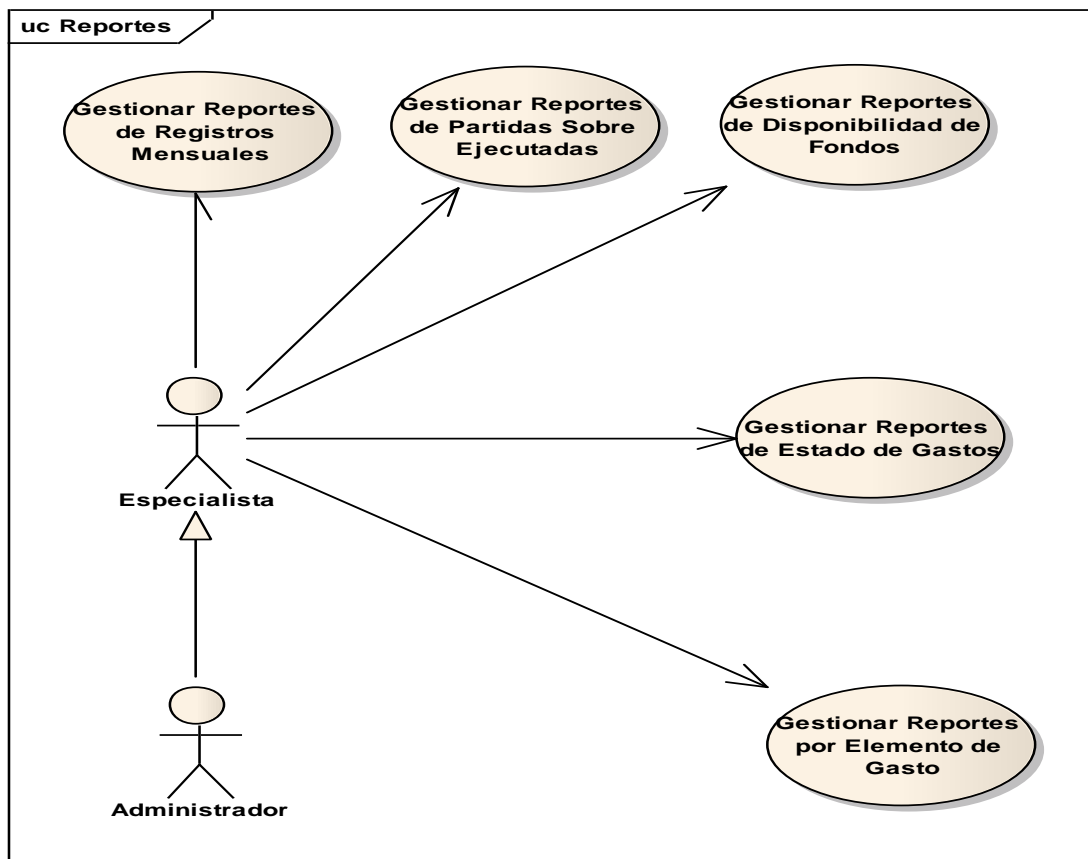


Figura 5. Diagrama de Casos de Usos Paquete de Reportes

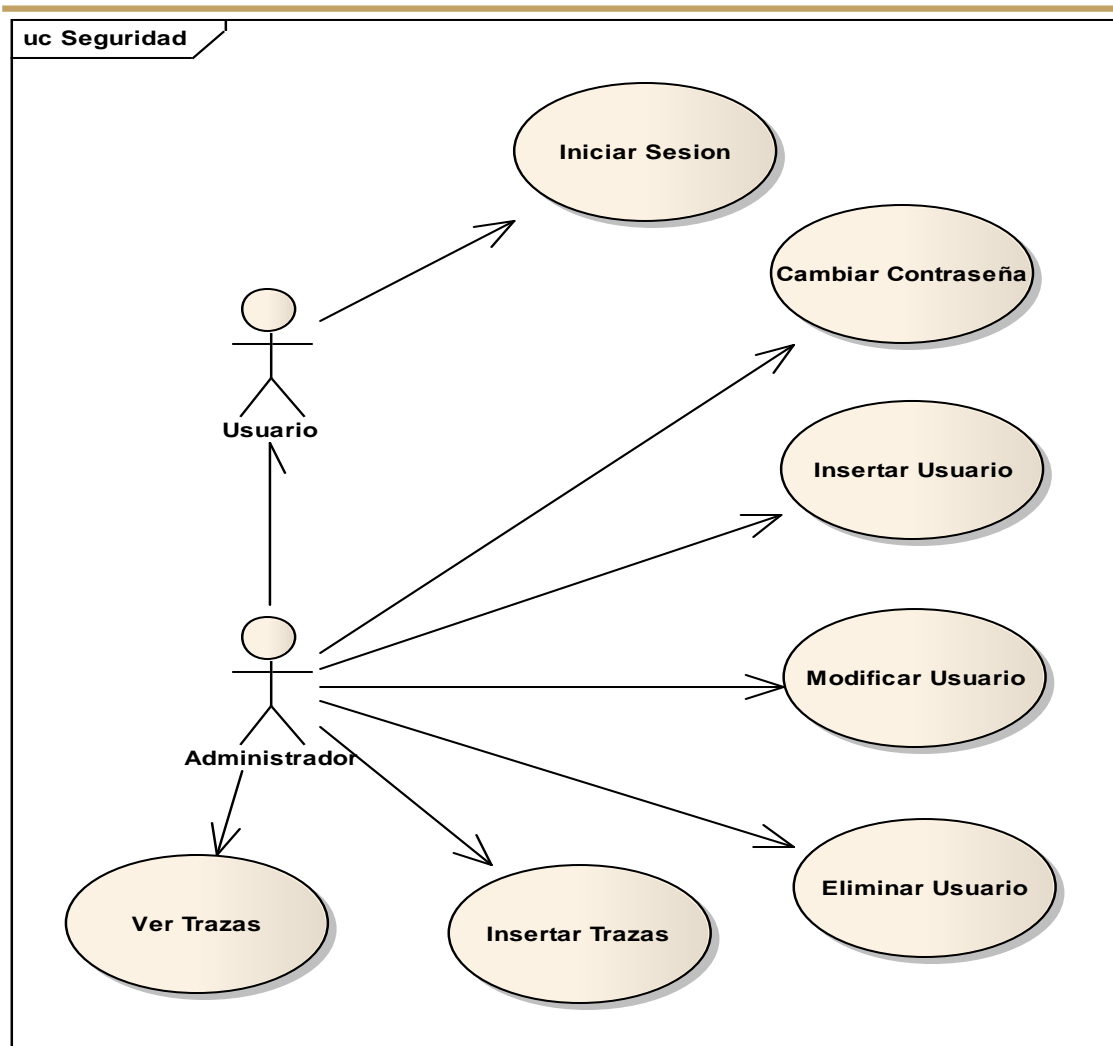


Figura 6. *Diagrama de Casos de Usos Paquete de Seguridad*

2.3. Análisis y diseño preliminar

El objetivo de la fase de análisis es la construcción de un sistema sin errores, mientras que la del diseño tiene como fin la construcción correcta de un sistema. El diseño preliminar constituye el paso intermedio entre el análisis y el diseño, y es precisamente el que posibilita que se puedan entender por completo los requerimientos, refinando y eliminando la ambigüedad de los mismos, a través del vínculo existente entre los casos de uso y los objetos del modelo del dominio [25].

Descripción textual de los principales casos de uso

Inmediatamente después de definir los diagramas de casos de usos y sus relaciones y con el fin de proporcionar el análisis y el diseño posterior se modelaron los casos de uso.

A diferencia de otras metodologías, como RUP, la descripción de los casos de uso en Iconix, se representan de forma tal que se capturen las operaciones del usuario y las respuestas del sistema. En la tabla 1, 2 y 3 se muestran las descripciones de los casos de uso “Insertar Plan”, “Modificar Registro” e “insertar Asignación”. La totalidad de las descripciones por cada caso de uso de los diagramas, se encuentran en el fichero eap, el cual las recoge así como los otros diagramas del proceso de desarrollo de Iconix.

Caso de Uso: Insertar Plan
<p>Curso Básico:</p> <p>El especialista accede a la Interfaz principal del sistema da clic en el menú Plan para Visualizar todos los planes, luego da clic en el botón Nuevo Plan, el sistema muestra la Interfaz Añadir Nuevo Plan, el Especialista llena los campos con los datos correspondientes y da clic en el botón Crear Nuevo Plan. El sistema Verifica la Validez de los datos a insertar. El Sistema Inserta los Datos.</p>
<p>Curso alterno:</p> <p>Campos vacíos:</p> <p>El Sistema Muestra el Mensaje de Notificación Campos Obligatorios y regresa a la Interfaz Añadir Plan.</p> <p>Añadir Plan:</p> <p>El especialista accede a la Interfaz Principal da clic en el menú Nuevo Plan para añadir un nuevo Plan</p>

Tabla 1. Descripción textual de Caso de Uso Insertar Plan

Caso de Uso: Modificar Registro
<p>Curso Básico:</p> <p>El Especialista escoge la opción Registros de Cheques dentro de la pantalla principal, luego muestra la pantalla correspondiente a Cheques, luego selecciona el Cheque a modificar y pasa a la pantalla editar Cheque mostrando los datos del mismo. El Especialista modifica los datos que desee cambiar y</p>

hace clic en el botón Guardar Cambios. El sistema valida que no haya campos vacíos o caracteres inadecuados y comprueba que no exista Cheques la Base de datos, almacena la información y muestra un mensaje de confirmación.

Curso alterno:

Valida datos: el Especialista selecciona el cheque a modificar, el sistema valida si el cheque está confirmado, si lo esta muestra solo la pestaña Editar, si el Cheque no está confirmado permite entrar a la pantalla modificar Cheque
 Valida datos: el Especialista introduce los datos en un formato incorrecto, deja campos en blanco. El sistema muestra un mensaje de error.

Tabla 2. Descripción textual de Caso de Uso Modificar Plan

Caso de Uso: Insertar Asignación

Curso Básico: El especialista accede a la Interfaz principal del sistema da clic en el menú Plan para Visualizar todas las Asignaciones, luego da clic en el botón Nueva Asignación, el sistema muestra la Interfaz Añadir Nueva Asignación, el Especialista llena los campos con los datos correspondientes y da clic en el botón Crear nueva Asignación. El sistema Verifica la Validez de los datos a insertar. El Sistema Inserta los Datos.

Curso alterno:

Campos Vacíos: El Sistema Muestra el Mensaje de Notificación Campos Obligatorios y regresa a la Interfaz Añadir Asignación.
 Valida datos: el Especialista introduce los datos en un formato incorrecto, deja campos en blanco. El sistema muestra un mensaje de error.

Tabla 3. Descripción textual de Caso de Uso Insertar Asignación

Diagramas de robustez

Para obtener el diseño del módulo se “convirtieron” los casos de uso en relaciones entre objetos. La técnica que propone la metodología ICONIX es el análisis de robustez, puente que une la descripción de los casos de uso y el diseño. Además, los diagramas de robustez son la vía para identificar el primer conjunto de objetos de cada caso de uso [26].

La descripción de los casos de uso y los diagramas de robustez se deben acoplar correctamente. Por ende, el texto de la descripción de los casos de uso y los objetos del diagrama de robustez deben tener una relación 1:1, es decir, las frases sustantivas se convierten en entidades e interfaces, mientras que los verbos se convierten en controladores de estos objetos [26].

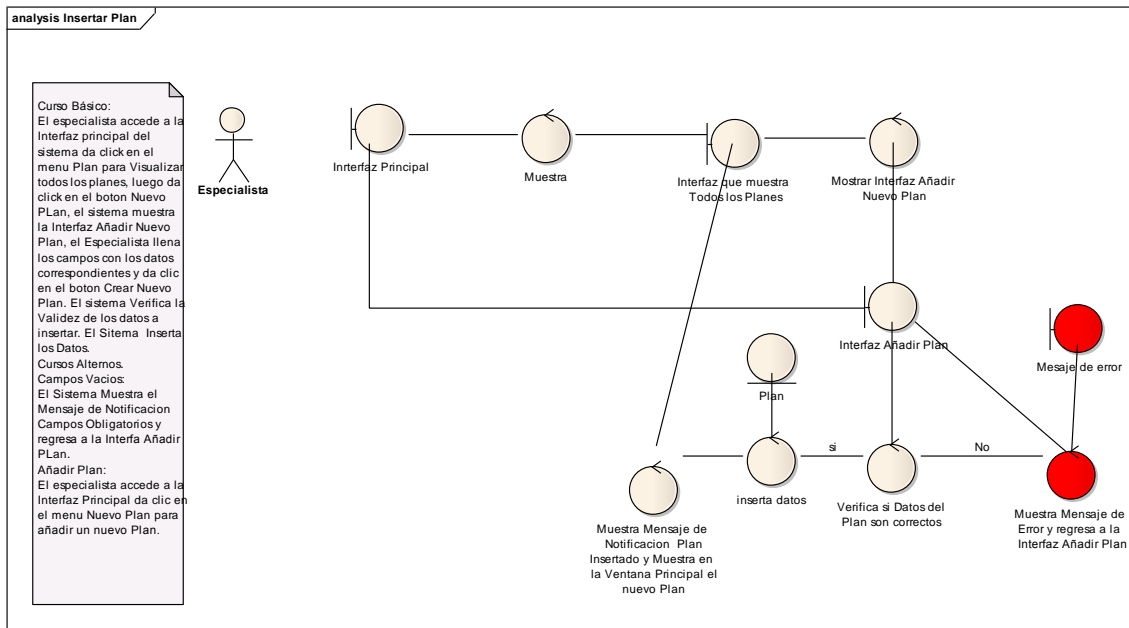


Figura 7. Diagrama de Robustez Insertar Plan

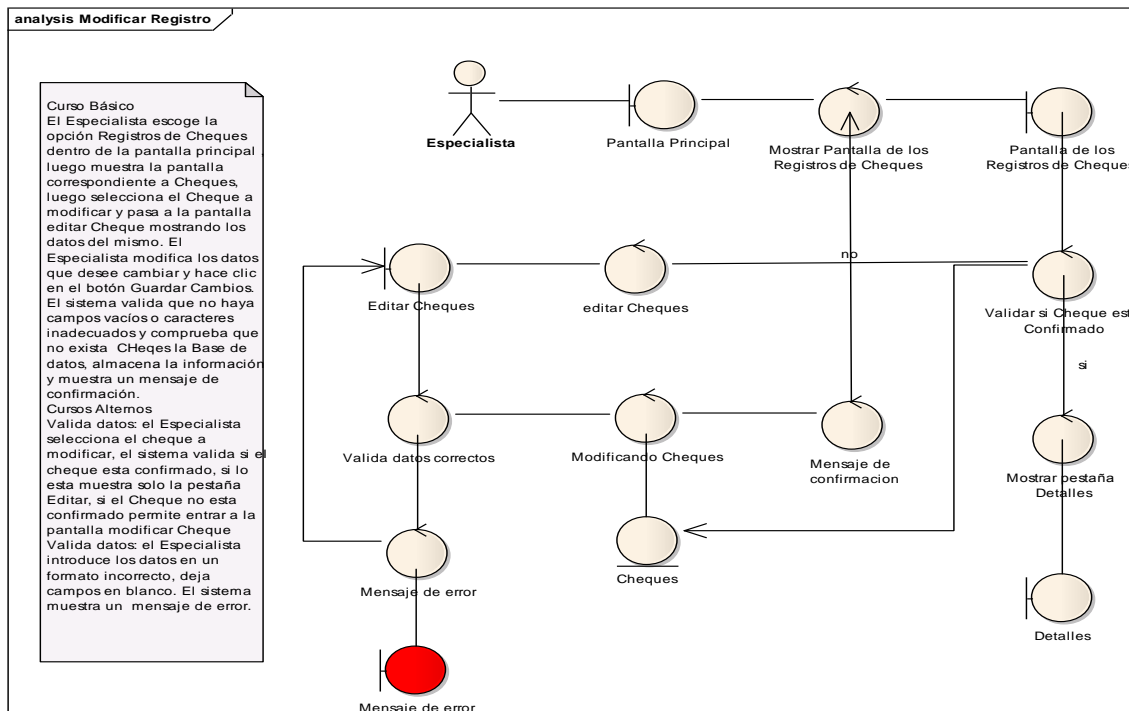


Figura 8. Diagrama de Robustez Modificar Registro

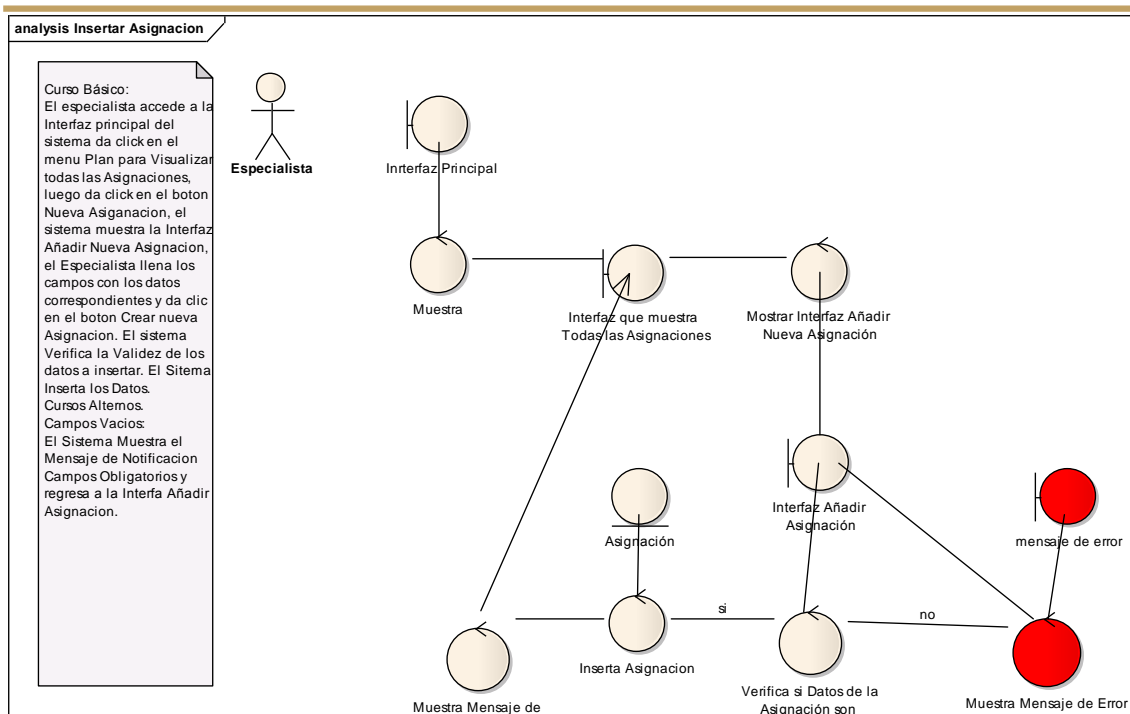


Figura 9. Diagrama de Robustez Insertar Asignación

Prototipación rápida

La Prototipación rápida se usa en Iconix para simular la interfaz final de la aplicación, para que el usuario tenga una idea clara del mismo y aporte su opinión y sugerencias. Los cambios al prototipo son planificados por los usuarios antes de la implementación y se espera que estos lo evalúen como si fuera el diseño final de la aplicación [35].

A continuación se muestran algunas de las interfaces del sistema.

Finanzas CUC

Iniciar sesión

Nombre de usuario:

Contraseña:

Iniciar sesión

Figura 10. Interfaz Iniciar Sesión

Bienvenido/a, Lia!

Sistema de Control de Finanzas CUC

1 elemento de gasto está sobre ejecutado ([Ver detalles](#)) ×

Figura 11. Interfaz Principal del Sistema

Disponibilidad de fondos 2014

Imprimir esta tabla

Elemento	Asignado	Ejecutado		Disponible	
Inversiones	0.00	0.00	0.00%	0.00	0.00%
Otros Servicios	0.00	0.00	0.00%	0.00	0.00%
Otros destinos	0.00	0.00	0.00%	0.00	0.00%
Gastos de Atencion al Hombre	0.00	0.00	0.00%	0.00	0.00%
Total Estimulacion en Divisas	0.00	0.00	0.00%	0.00	0.00%
Consumo Productivo	500.00	400.00	80.00%	100.00	20.00%
Servicios Recividos de Residentes	0.00	400.00	400.00%	(400.00)	(400.00%)
Atencion a Delegaciones Extranjeras	0.00	0.00	0.00%	0.00	0.00%
Otros Destinos no Nominalizados	0.00	0.00	0.00%	0.00	0.00%
Inventario Inicial	0.00	0.00	0.00%	0.00	0.00%
Inventario Final	0.00	0.00	0.00%	0.00	0.00%
Cuentas por pagar al inicio	0.00	0.00	0.00%	0.00	0.00%
Cuentas por pagar al final	0.00	0.00	0.00%	0.00	0.00%
Demanda de Compras de Divisas	0.00	0.00	0.00%	0.00	0.00%
Total	500.00	800.00	160.00%	(300.00)	(60.00%)

Figura 12. Interfaz reporte de Disponibilidad de Fondos

Modelo de despliegue

Dentro de la arquitectura técnica, un paso muy importante es el diagrama de despliegue, el cual representa las principales capas que conforman la aplicación, siendo un modelo de objetos que describe la distribución física del sistema, además permite distribuir las funcionalidades entre nodos. Conjuntamente es una colección de nodos y arcos, donde cada nodo representa un recurso de cómputo, normalmente un procesador o un dispositivo de hardware similar. Ver Figura 13.

En el desarrollo del módulo propuesto se tuvieron en cuenta los principios de una arquitectura multicapa, lo cual aumenta la mantenibilidad y flexibilidad del sistema, pues a la hora de corregir, perfeccionar o adaptar el software resulta sencillo detectar en qué lugar han de realizarse las modificaciones necesarias.

En el servidor se aloja el servidor Web Apache y el sistema gestor de base de datos MySQL con la base de datos correspondiente para almacenar, consultar y actualizar la información por los usuarios. La computadora cliente juega el rol del conjunto de computadoras destinadas al proceso para gestionar la información ubicada en el servidor principal. La comunicación que se establece entre el servidor principal y las computadoras clientes es a través de un conjunto de protocolos HTTP.

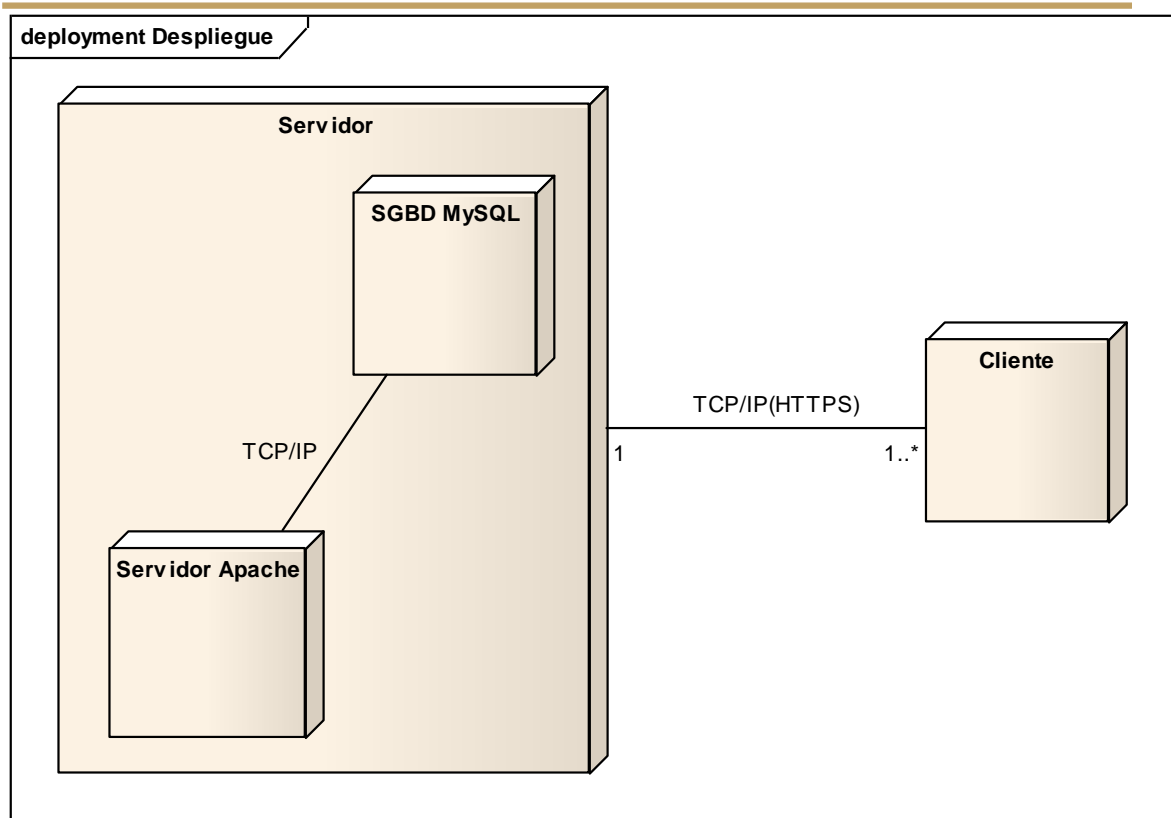


Figura 13. *Modelo de Despliegue*

2.4. Diseño detallado

El análisis y diseño preliminar solo era una teoría de cómo funcionarían las clases de la aplicación, al contrario que con el diseño detallado se logra asignar funciones a cada una de las clases que fueron detectadas. Además es donde se empiezan a ver qué métodos llevarán las clases del sistema. Esto se debe que hasta ahora solo se ha interactuando con los objetos de las clases con los actores y con otros objetos de manera dinámica, por lo que se tiene suficiente información como para poder empezar a especificar los métodos de las respectivas clases.

Diagramas de secuencia

Siguiendo el procedimiento de la metodología ICONIX, luego de haber representado el diagrama de robustez para cada caso de uso y de finalizar la etapa de análisis con el desarrollo de la arquitectura técnica, se está en condiciones de realizar la etapa de diseño detallado, en ella se realiza el diagrama de secuencia. Éste es un punto importante, pues es el encargado de diseñar la interacción de cada caso de uso, además de conocer los métodos

que componen cada clase; por lo que ICONIX propone poner un diagrama para cada caso de uso, lo que facilita entender su diseño.

El diagrama de secuencia tiene 10 objetivos elementales, entre ellos los más importantes son:

- Asignar comportamiento a las clases.
- Mostrar en detalle como las clases interactúan entre sí durante el tiempo de vida del caso de uso.
- Terminar la distribución de las operaciones entre clases.

En la Figura se muestra el Diagrama de Secuencia antes mencionado.

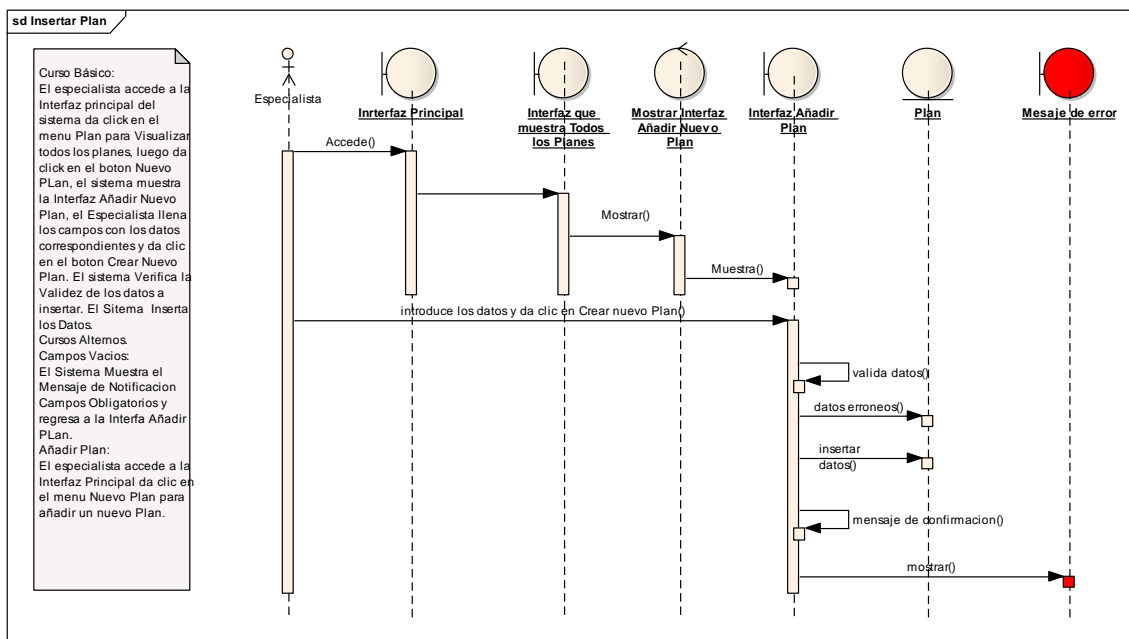


Figura 14. Diagrama de Secuencia Insertar Plan

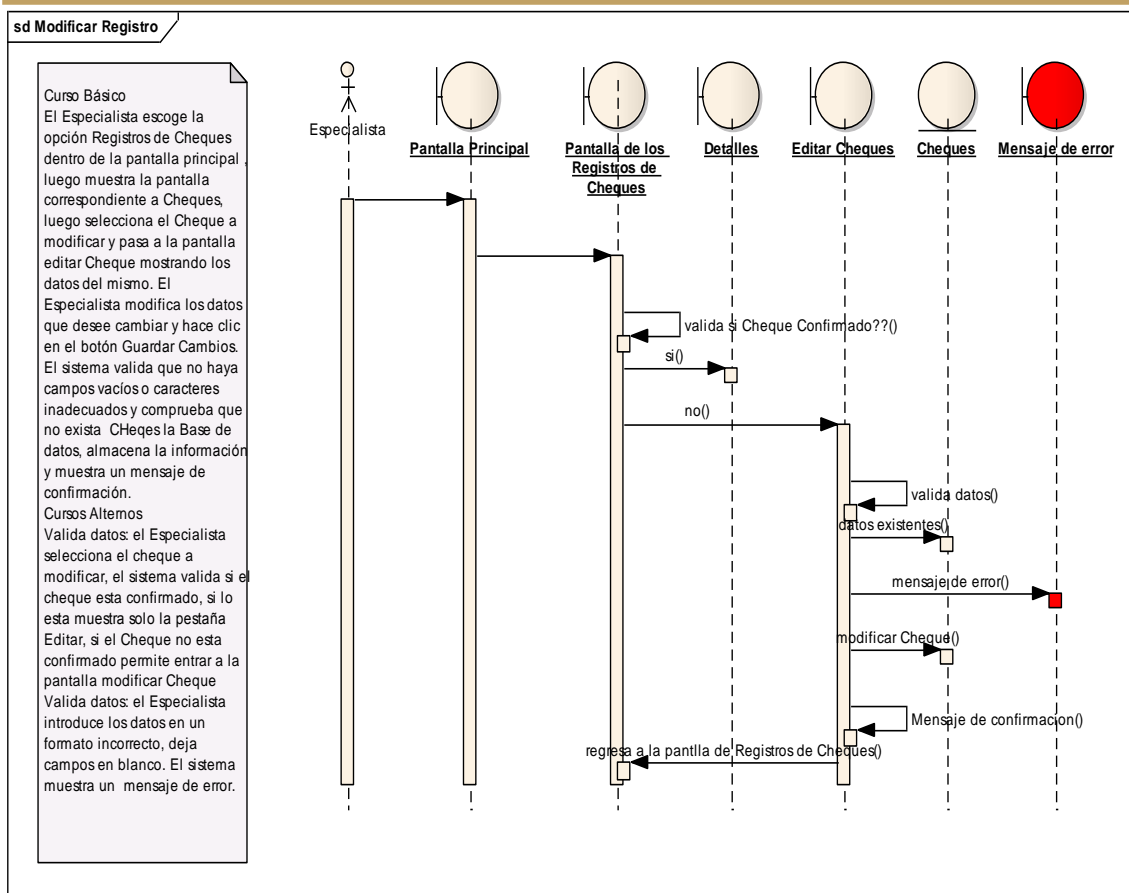


Figura 15. Diagrama de Secuencia Modificar Registro

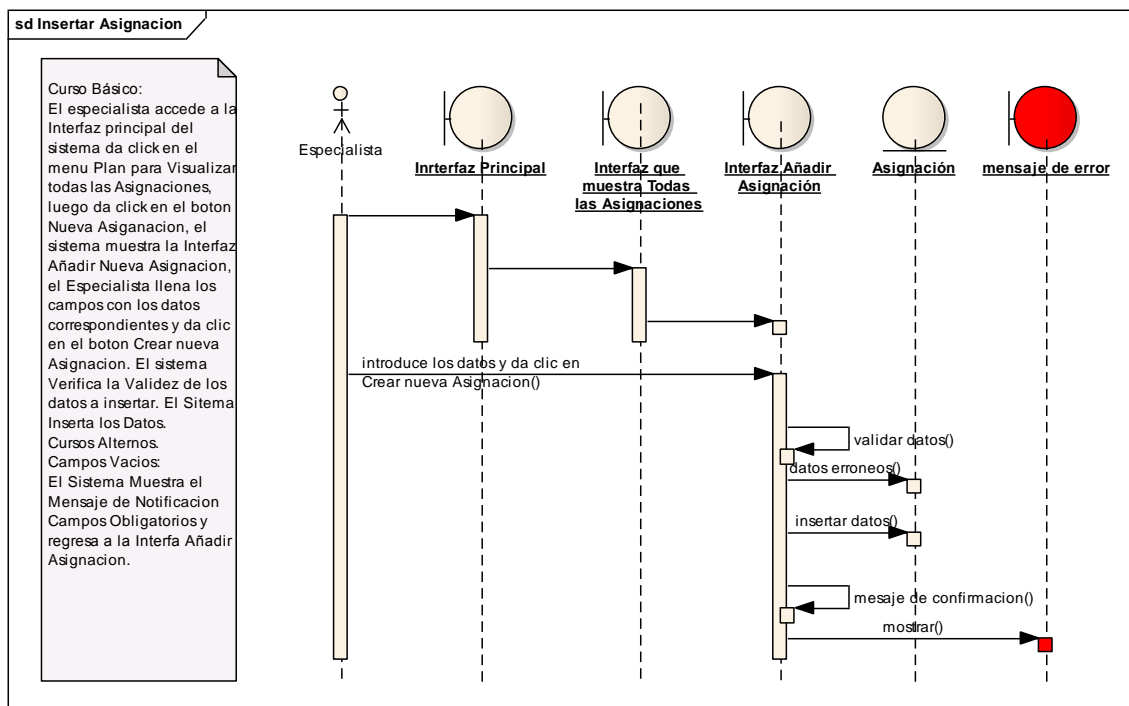


Figura 16. Diagrama de Secuencia Insertar Asignación

Diagrama de Clases

A partir de la modelación de los diagramas de secuencia, se pudo refinar los diagramas de robustez y modelo del dominio, a raíz de ello se desarrollaron los principales métodos y atributos que lo componen, dando lugar al diagrama de clase. Este último tiene como principal objetivo alcanzar una comprensión más precisa de los requisitos que se describieron en la primera fase de desarrollo de la metodología, así como lograr darle vida a los requisitos en el sistema, razonar los aspectos internos y la flexibilidad ante los cambios de la reutilización [27].

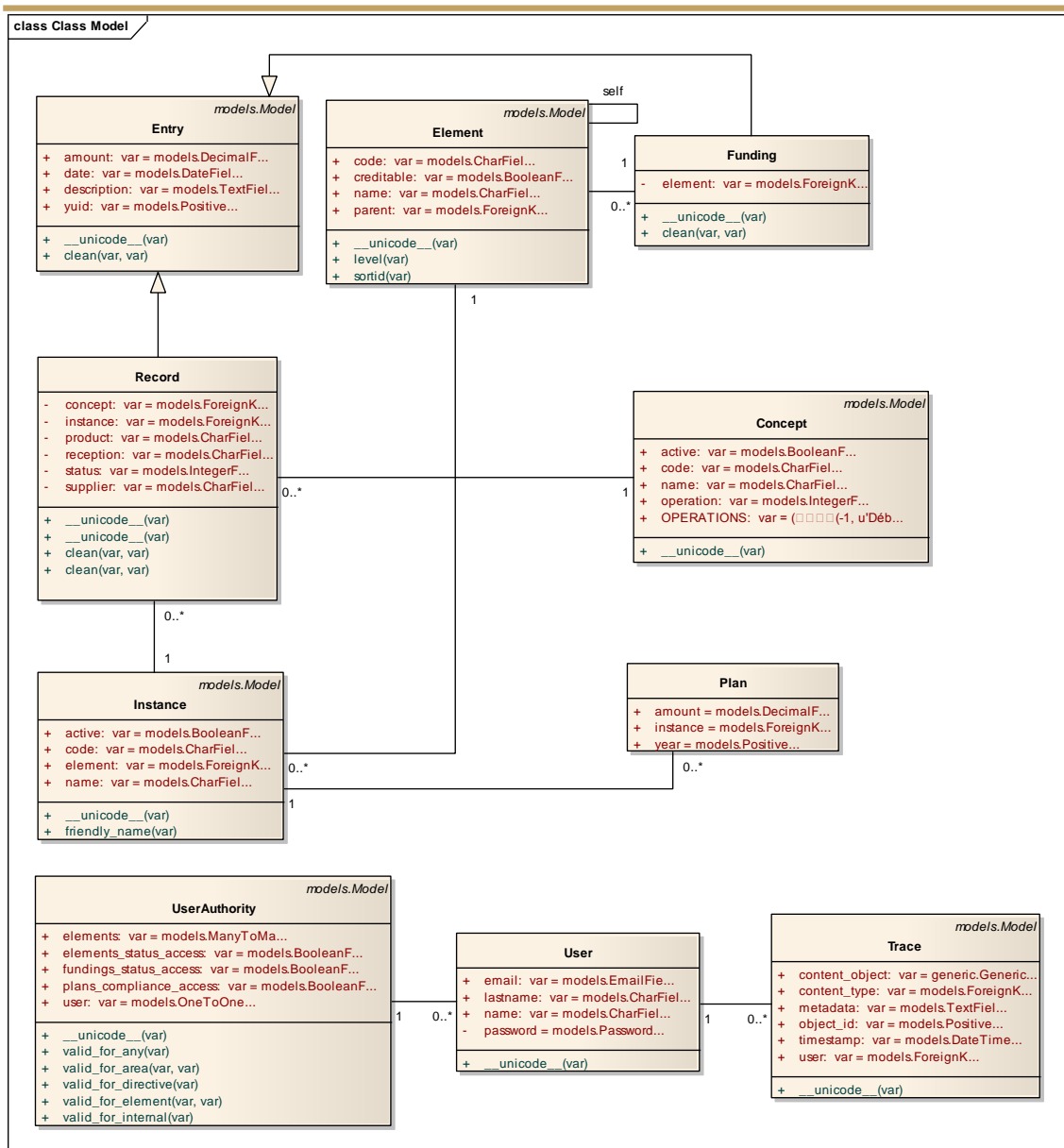


Figura 17. Diagrama de Clases

Diseño de la base de datos

El modelo de datos se ha convertido en el diagrama común para los desarrolladores de aplicaciones y desarrolladores de bases de datos. A partir del diagrama de clases como modelo lógico de la base de datos se puede obtener el modelo de datos; en este modelo se define las estructuras de datos que se utilizan en el sistema y describe la estructura lógica de la información que queda almacenada en la base de datos [39]. El propio flujo de trabajo del análisis y el diseño permitió definir el modelo lógico de datos y a partir de este confeccionar el modelo de datos (Ver Anexo 4).

2.5. Implementación

Una vez concluido el diseño detallado, a través de los diagramas de secuencia de cada caso de uso, se procedió a traducir todo el diseño a código fuente. Esta fase posibilita definir la organización y estructura del código, e implementar las clases que se identificaron durante el diseño.

Estándar de codificación

En el desarrollo de la aplicación se establecieron estándares de código para facilitarle y agilizarle tanto a los desarrolladores de la aplicación como a los responsables del mantenimiento y actualización la lectura del código.

A continuación se muestran algunos de los estándares de código establecidos:

Comentarios

Los comentarios están escritos en español, siendo frases completas y comenzando con letra mayúscula.

Variables

Los nombres de las variables son escritos en minúsculas y las palabras se separan con guiones bajos (_).

Definición de clases y funciones

En la definición de las clases y las funciones no se hace uso de abreviaturas, siendo lo más descriptivo posible y comenzando el nombre siempre con mayúscula. En el caso de que los nombres sean muy largos son separados por guiones bajos.

2.6. Prueba

Iconix plantea que la fase de prueba debería comenzarse antes de la implementación. La preparación para la prueba comienza desde la etapa de análisis, donde se identifican los casos de prueba mediante el uso de los diagramas de robustez, los cuales son codificados durante la implementación. La realización de la prueba en etapas tempranas hace posible eliminar gran cantidad de errores, incluso antes de que existan [24].

Las pruebas están estrechamente relacionadas con los requerimientos, ya que se “prueba” que los mismos se satisfagan. Debe haber al menos una prueba que asegure que cada requerimiento ha sido implementado correctamente. Para ello Iconix propone algunos tipos de prueba y cuándo usarlos [24].

La prueba de caja negra se refiere a las pruebas que se llevan a cabo sobre la interfaz del software. O sea, los casos de prueba pretenden demostrar que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se produce un resultado correcto, así como que la integridad de la información externa se mantiene.

Se probaron las tecnologías utilizadas antes y luego de la integración. Se realizaron varias pruebas para cada caso de uso, verificando que se diera cumplimiento a cada requerimiento del sistema, para lo que se utilizaron datos reales.

Un caso de prueba es un conjunto de entradas de pruebas, condiciones de ejecución y resultados esperados desarrollados para cumplir un objetivo en particular o una función esperada. La entidad más simple que siempre es ejecutada como una unidad, desde el comienzo hasta el final.

Caso de uso	Buscar Plan
Caso de prueba	Encontrar planes que pertenecen a un indicador especificado
Entrada	El especialista necesita la relación de planes pertenecientes al Indicador especificado. El indicador especificado no tiene planes asignados
Resultado	Se muestra un mensaje al especialista informándole que “No hay planes que coincidan con su búsqueda”.

Tabla 4. Estructura de los casos de prueba

2.7. Valoración de sostenibilidad

La valoración de la sostenibilidad de un Producto Informático se entiende como el proceso de evaluación de impactos ambientales, socio humanistas, administrativos y tecnológicos de dicho producto, previsibles desde el diseño del proyecto, que favorece su autorregulación, para la satisfacción de la necesidad que resuelve, con un uso racional de recursos y la toma de decisiones adecuadas a las condiciones del contexto y el cliente [28].

Precisamente es la valoración de sostenibilidad la que asegura la efectividad del sistema informático como solución propuesta al problema planteado, garantizando que satisfaga la necesidad que lo origina y las expectativas del

cliente. Es por ello que se tuvo en cuenta que este fuera sostenible desde el punto de vista: administrativo, socio-humanista, ambiental y tecnológico.

Dimensión administrativa

Esta dimensión se orienta en el ahorro, gasto, calidad de la producción y los servicios, administración de recursos y en la toma de decisiones administrativas que brinda como resultado el producto informático.

El método empleado para determinar el esfuerzo requerido para el desarrollo del software fue el de estimación por Puntos de Casos de Uso, mediante una aproximación a los primeros casos de usos desarrollados. Este método resulta efectivo para este proyecto pues la metodología sustenta su desarrollo en iteraciones continuas donde los primeros casos de uso son los que permiten diseñar la mayor parte de la arquitectura del software y los que a su vez ayudan a mitigar los riesgos más significativos.

El cálculo del esfuerzo arrojó un valor de \$ 4752 como se muestra en la tabla 12 (ver anexo 5) el cual se considera apropiado según la envergadura y el alcance del proyecto. El valor está dado por el tiempo dedicado a este el cual disminuyó considerablemente como resultado del empleo de Framework Django el cual aporta un ciclo de desarrollo muy ágil.

La inversión en recursos informáticos es mínima pues la infraestructura que sirve de soporte al proyecto ya estaba creada posibilitando no incurrir en gastos. El sistema más que minimizar la mano de obra la optimiza. Aporta mejoras significativas para la gestión de la información en el proceso de control del Plan de la Economía disminuyendo el trabajo manual de los especialistas y favoreciendo el flujo de trabajo entre las partes involucradas. Proporciona además una disminución considerable en los tiempos de búsqueda y garantiza la perdurabilidad de información valiosa. No se incurren en gastos de electricidad asociado al uso de las computadoras y el servidor porque esta situación ya se presentaba.

Las herramientas utilizadas para el desarrollo del software en su totalidad son libres, con lo que se exonera a la institución (Universidad de Holguín “Oscar Lucero Moya”) de incurrir en gastos para llevar a cabo el desarrollo del proyecto.

A partir de lo analizado y los beneficios que proporciona el software, se arribó a la conclusión que éste es sostenible desde la dimensión administrativa.

Dimensión socio-humanista

El sistema resuelve la necesidad social de humanizar el trabajo ya que permitirá que el usuario se sienta cómodo, mediante una interfaz sencilla y que tenga reconocimiento visual. La facilidad del uso del sistema, unido a la rapidez con que puede realizar las operaciones constituyen comodidades que previenen además padecimientos propios del uso de las computadoras, que se disminuyen estando frente a las mismas en una posición correcta y usando protectores de pantalla y no estando mucho tiempo delante del ordenador.

El sistema se puede implantar en otras universidades del país para la gestión del presupuesto en divisas, pues la necesidad que hace que surja el mismo puede no ser exclusiva de la Universidad de Holguín “Oscar Lucero Moya” de Holguín, sino en cualquier institución análoga del país donde se presenten las mismas deficiencias que son erradicadas con el mismo. También el desarrollador o el que implanta el sistema debe favorecer la aceptación del mismo mediante una ayuda como es el manual de usuario.

El software no genera empleos con su uso. Además de contribuir a la conciencia del desarrollador para que desarrolle un sistema robusto a prueba de fallas y con la mayor calidad. Partiendo de las características expuestas se puede afirmar que el sistema es sostenible en esta dimensión.

Dimensión ambiental

En esta dimensión se valoran las condiciones favorables o no para los usuarios del sistema y la forma en que minimiza daños e impactos negativos. El producto informático contribuirá a la conservación del Medio Ambiente, especialmente con el ahorro de material de oficina (hojas, lápices y espacio para los archivos), que se obtienen de la naturaleza y que se fabrican industrialmente y evitando la tala indiscriminada de árboles.

El diseño del sistema contempla las exigencias fisiológicas del ser humano al seleccionar el tamaño de letra, espaciamiento entre caracteres, cumple con el uso de colores agradables a la vista, propone una interfaz sencilla y un diseño similar en todas las interfaces de navegación, lo cual, conjuntamente con el manual de usuarios, favorece su aceptación por parte de los usuarios.

A partir de lo analizado y plasmado anteriormente, se arribó a la conclusión que el sistema es sostenible desde la dimensión ambiental.

Dimensión tecnológica

Al implantar el sistema, los usuarios no mostrarán resistencia a este cambio ya que son personas que mantienen contacto con las computadoras. Para el uso y explotación del sistema los usuarios recibirán una breve capacitación teniendo en cuenta que el sistema es una aproximación acertada del trabajo que se realiza manualmente, además contarán con un manual de usuario como material complementario para el trabajo con el software.

Las características que poseen las computadoras utilizadas por los trabajadores cumplen con los requerimientos necesarios para hacer uso del sistema y además se encuentran conectadas a la red, por otra parte, los cambios tecnológicos que se puedan producir en el futuro para lograr la optimización y mejora de los procesos realizados repercutirá en el producto informático que los automatice; en el caso de este sistema se prevé que evolucione en el tiempo, debido a la flexibilidad que proporciona.

A partir de lo analizado y plasmado anteriormente, se arribó a la conclusión que el sistema es sostenible desde la dimensión tecnológica. El sistema puede ser reutilizado ya que en su código fuente las funcionalidades se programaron de forma general que sirven para cada caso particular

Conclusiones de la valoración de sostenibilidad

El sistema es sostenible en todas las dimensiones, analizadas, pues existe una coherencia entre la necesidad social o el problema que generó la elaboración del mismo y los recursos empleados para solucionarlo. Además las herramientas propuestas para dar solución al sistema propuesto cumplen las expectativas del desarrollador cumpliendo con los requerimientos planteados, reafirmando que estamos en presencia de un producto sostenible.

2.8. Valoración de la Propuesta de Solución

Una vez concluido el sistema, se procedió a la elaboración de encuestas, a través del criterio de expertos, con el objetivo de evaluar la satisfacción de los usuarios finales y obtener valoraciones conclusivas del sistema, mediante el método Delphi.

El método Delphi considerado como uno de los métodos subjetivos de pronósticos más confiables, constituye un procedimiento para confeccionar un cuadro de la evolución estadística de las opiniones de expertos o usuarios en un tema tratado; también es la utilización sistemática del juicio intuitivo de estos expertos para obtener un consenso de opiniones sobre la posible aplicabilidad de una propuesta determinada [29].

Para la selección de los expertos, se consultaron profesionales con experiencia, creatividad, capacidad de análisis y de pensamiento en el tema tratado. Se tuvo en cuenta, además, la disposición de éstos para responder la encuesta elaborada, mostrada en el Anexo 7, para obtener a partir del procesamiento de la misma el coeficiente de competencia. Determinando este coeficiente, a partir de la opinión del experto sobre su nivel de conocimiento, acerca del problema que se está resolviendo y con las fuentes que le permiten argumentar sus criterios.

Se escogieron sólo los de alta (6) y mediana (4) competencia para un total de 6 expertos seleccionados de los 10 encuestados, los cuales poseen preparación en el tema y conocimientos en las tecnologías informáticas, con una experiencia mínima de 1 año.

Tras realizar las encuestas a los expertos la segunda ronda fue procesada por el método Delphi, ver Anexo 8, con el propósito de encontrar un consenso entre los encuestados.

Los aspectos de evaluación del sistema se muestran a continuación:

1. ¿La aplicación Web favorece la gestión de la información del Plan de la Economía de la Universidad de Holguín “Oscar Lucero Moya”?
2. ¿Cómo valora la eficiencia del sistema con respecto a los procesos que se realizaban de forma manual anteriormente?
3. ¿Qué le parece el uso de los colores e imágenes del sistema?
4. ¿Qué tan “amigable” le resultó el diseño de las interfaces del sistema?
5. ¿Cómo considera usted el acceso a las funcionalidades del sistema?
6. ¿Cómo valora el uso que se le puede dar al sistema?

A partir del procesamiento estadístico de los aspectos anteriormente mencionados y tratados en la encuesta, se determinó, como resultado final de la evaluación, que los encuestados coinciden en que todos los aspectos son

“Muy Adecuados”, esto se puede apreciar en el Anexo 9, lo cual influye positivamente en la calidad del sistema. Se puede concluir con los resultados obtenidos que el grado de satisfacción es “Alto”, lo cual permitió percibir la satisfacción de las necesidades iniciales.

Conclusiones parciales del Capítulo

En este capítulo se determinaron, a través de la metodología ICONIX, los objetos o entidades fundamentales implicadas en el dominio del problema y la relación existente entre ellas. Se identificaron las exigencias que debe cumplir la solución propuesta, así como los usuarios finales y las acciones que estos deben ejecutar. Además, se especificaron los cursos básicos y alternos de las acciones del sistema, así como los procesos de diseño, implementación y prueba de la metodología utilizada. Se realizó un estudio de la sostenibilidad administrativa, socio-humanista, ambiental y tecnológica del sistema, por lo que se arribó a la conclusión de que su desarrollo es factible y sostenible.

Conclusiones generales

1. El producto informático desarrollado favorece la Gestión de la información en el área de Finanzas CUC de la Universidad de Holguín "Oscar Lucero Moya", cumpliendo con el objetivo trazado en la investigación.
2. La metodología de desarrollo de software utilizada para realizar el diagnóstico del problema y modelar la solución, resultó fundamental para garantizar la calidad en los procesos de ingeniería de software implicados.
3. Python como lenguaje de programación y MySQL como sistema gestor de base de datos constituyeron tecnologías muy adecuadas para el desarrollo del sistema.
4. Al valorar las dimensiones, administrativas, socio-humanistas, ambiental y tecnológica se puede afirmar que el producto es sostenible.

Recomendaciones

A partir de los resultados obtenidos y de la investigación realizada se propone:

- Implementar funcionalidades que permitan analizar de modo gráfico el estado de los distintos indicadores del plan de la economía.
- Implementar funcionalidades que permitan exportar los reportes a hojas de cálculo de Excel y PDF.



Referencias bibliográficas

1. Fayol H, *Administración General e Industrial*. 1969, Buenos Aires Argentina.
2. Almaguer R. *Introducción del concepto de calidad total en la actividad contable*. 2008; Available from: http://www.betsime.disaic.cu/secciones/fin_so_03.htm.
3. Benítez M, M.M., *Contabilidad y finanzas para la formación económica de los cuadros de dirección*. 1997, La Habana: Pueblo y Educación.
4. Berberena, R.P. *Estrategia y dirección estratégica*. 2010; Available from: <http://www.gestiopolis.com/administracion-estrategia-2/sistemas-integrados-gestion-informacion.htm>.
5. *El control y la gestión estratégica de las organizaciones* 2006; Available from: http://www.elprisma.com/apuntes/administracion_de_empresas/controlstrategico.
6. Managerial, S.A. *Managerial Analyzer: Managerial Analyzer Software*. 2005; Available from: <http://www.managerialanalyzer.com/web/index.php>.
7. Analysis, F. *Finaplan*. 2011; Available from: <http://www.finanplan.com/aplicaciones-informaticas/fin-analysis>
8. MIC. *Software Contables-Financieros: Agencia de Supervisión y Control*. 2011; Available from: http://www.acs-mic.cu/sw_certificados.htm.
9. Blanco, L., *La informática en la dirección de empresas*. 2011, La Habana: Félix Varela.
10. Fernández, I., *Contar con tecnología*. Granma, 2012.
11. RODRÍGUEZ, H., *Cuba avanza en la implementación del software libre*. 2012.
12. *Introducción a Patrones*. 2002; Available from: <http://www.mcc.unam.mx/~cursos/Algoritmos/javaDC99-2/patrones.html>.
13. E. Gamma, R.H., R. Johnson y J. Vlissides, ed. *Design Patterns Elements of Reusable Object-Oriented Software*. 1995: Washington.
14. *Sistemas gestores de bases de datos*. 2007; Available from: <http://www.desarrolloweb.com/articulos/sistemas-gestores-bases-datos.html>.
15. *Qué es Javascript*. 2001; Available from: <http://www.desarrolloweb.com/articulos/25.php>.
16. Guido van Rossum Fred L. Drake, J., *Guía de aprendizaje de Python Release 2.0*. 2000.
17. *Symfony en pocas palabras*. 2007; Available from: http://www.librosweb.es/symfony_1_0/capitulo1/symfony_en_pocas_palabras.html
18. Zarete, S., *Django el Framework Web Definitivo*. 2008.
19. Adrian Holovaty, J.K.-M., *El libro de Django*. 2008.
20. Patón, M.y.F., Eduardo. *Metodologías de Desarrollo software*. 2002; Available from: <http://alarcos.inf-cr.uclm.es/doc/ISOFTWAREI>.
21. Rosenberg, D., & Scott, K, *Applying Use Case Driven Object Modeling with UML* 2001: Addison Wesley.
22. *Enterprise Architect - Herramienta de diseño UML*. 2000-2007; Available from: <http://www.sparxsystems.com.ar/products/ea.html>.
23. Cornejo, J.E.G. *Lenguaje de Modelado Unificado*. 2011; Available from: <http://www.docirs.cl/uml.htm>.
24. Rosenberg, D.a.M.S., *Use Case Driven Object Modeling with UML.Theory and Practice*. 2007.
25. Rodríguez C, T.M., *Sistema de Gestión de Información Editorial. Semanario Ahora*, 2008.



26. DOUG ROSENBERG, M.S., AND MARK COLLINS-COPE, *Agile Development with ICONIX Process—People, Process, and Pragmatism*. 2005.
27. Jacobson, I., Rumbaugh, James and Booch, Grady. , *El Lenguaje Unificado de Modelado. Manual de Referencia*. 2000: Addison Wesley.
28. Concepción, R., *Procedimiento para la valoración de sostenibilidad de un Producto Informático*. 2006.
29. García, D.F.R.E.D.R.C., *El Método Delphy para el Procesamiento de los Resultados de Encuestas a Expertos o Usuarios en Estudios de Mercado y en la Investigación Educativa*.

Bibliografía

- Software Libre.). From <http://www.hispalinux.es>
- Enterprise Architect - Herramienta de diseño UML. (2000-2007). from <http://www.sparxsystems.com.ar/products/ea.html>
- Qué es Javascript. (2001). from <http://www.desarrolloweb.com/articulos/25.php>
- Introducción a Patrones. (2002). from <http://www.mcc.unam.mx/~cursos/Algoritmos/javaDC99-2/patrones.html>
- Autores, C.D. PHP Manual. Manual de Usuario. (2004). from <http://www.php.net/docs.php>
- Introducción a PostgreSQL. (2004). from http://www.geocities.ws/fodcl/doc_post/doc_post_histo.html
- El control y la gestión estratégica de las organizaciones (2006). From http://www.elprisma.com/apuntes/administracion_de_empresas/controlestrategico
- Sistemas gestores de bases de datos. (2007). from <http://www.desarrolloweb.com/articulos/sistemas-gestores-bases-datos.html>
- Symfony en pocas palabras. (2007). from http://www.librosweb.es/symfony_1_0/capítulo1/symfony_en_pocas_palabras.html
- El lenguaje de programación Java. (2010). from <http://www.lenguajes-de-programacion.com/programacion-java>
- La Definición de Software Libre. (2010). Retrieved 21-01-10, 2010, from <http://www.gnu.org/philosophy/free-sw.es.html>
- MySQL Web Site. (2011). from <http://www.mysql.com>
- Adrian Holovaty, J.K.-M. (2008). *El libro de Django*.
- Almaguer R. (2008). Introducción del concepto de calidad total en la actividad contable. From http://www.betsime.disaic.cu/secciones/fin_so_03.htm
- Alvarez, Miguel Angel. (2003). Qué es Python. Retrieved 28-02-11, 2011, from <http://www.desarrolloweb.com/articulos/1325.php>
- Alvarez, S.). Arquitectura Cliente_serv. Retrieved 22-10-10, 2010, from http://www.serv\Arquitectura_cliente-servidor.html
- Analysis, Fin. (2011). Finaplan. From <http://www.finanplan.com/aplicaciones-informaticas/fin-analysis>
- Barahona (2002) ¿Qué es el software libre? .
- Benítez M, Miranda M. (1997). *Contabilidad y finanzas para la formación*



- económica de los cuadros de dirección*. La Habana: Pueblo y Educación.
- Berberena, R.P. (2010). Estrategia y dirección estratégica. From <http://www.gestiopolis.com/administracion-estrategia-2/sistemas-integrados-gestion-informacion.htm>
- Blanco, L. (2011). *La informática en la dirección de empresas*. La Habana: Félix Varela.
- Booch, G.R., J. & Jacobson (1999). *El Lenguaje Unificado de Modelado* (3 ed.): Addison-Wesley.
- Brea, Roberto Castro. (2010). *Edición de perfiles de tubería en el diseño de redes hidráulicas de abasto en la EIPH Holguín "Raudal": Módulo de un Sistema CAD distribuido*. Oscar Lucero Moya, Holguín.
- Canos, Jose H. (2008). Metodologías Ágiles en el Desarrollo de Software. *Universidad Politécnica de Valencia*.
- Concepción, R. (2006). *Procedimiento para la valoración de sostenibilidad de un Producto Informático*.
- Cornejo, J. E. G. (2011). Lenguaje de Modelado Unificado. From <http://www.docirs.cl/uml.htm>
- DOUG ROSENBERG, MATT STEPHENS, AND MARK COLLINS-COPE. (2005). *Agile Development with ICONIX Process—People, Process, and Pragmatism*.
- Duvallier. (2010). Aptana uno de los mejores IDE multiplataforma. Retrieved 20-02-11, 2011, from <http://www.aclibre.org/archives/Aptana/Aptana.html>
- E. Gamma, R.H., R. Johnson y J. Vlissides (Ed.). (1995). *Design Patterns Elements of Reusable Object-Oriented Software*. Washington.
- Fabiano, Carlos. (2010). Enterprise Architect 7.1 Full español. Retrieved 25-03-11, 2011, from http://www.taringa.net/posts/taringa/7465662/Enterprise-Architect-7_1-Full-espanol.html
- Fayol H. (1969). *Administración General e Industrial*. Buenos Aires Argentina.
- Fernández, I. (2012). Contar con tecnología. *Granma*.
- G Booch, J Rumbaugh, I Jacobson. . (1999). *The Unified Software Development*. Boston, US: Addison-Wesley.
- García, Ing. Iraimi Bermúdez. (2009). *Diseño e implementación del sistema de información al cliente en el destino turístico Holguín*. (Master), Holguín.
- García, Jesús Espino. (2008). Django: Framework MVC en python. Retrieved 15-

- 03-11, 2011, from <http://www.whyfloss.com/en/conference/madrid08/getpdf/>
- Gosling J, McGilton H. (1996). White Paper. The Java Language Environment. From <http://java.sun.com/docs/white/langenv/index.html>
- Guido van Rossum Fred L. Drake, Jr. (2000). *Guía de aprendizaje de Python Release 2.0*.
- Gutierrez, J. (2008). "Qué es un framework".
- Ilmaz, M. (2006). Charla: Desarrollo web en Python usando el framework Django. Retrieved 23-01-1011, 2011, from <http://the-geek.org/django-book/chapter1/>
- Jacobson, I., Rumbaugh, James and Booch, Grady. . (2000). *El Lenguaje Unificado de Modelado. Manual de Referencia*: Addison Wesley.
- Johnson, R. (2009). *Evolving Frameworks: A Pattern Language for Developing Object-Oriented Frameworks*.
- Loaiza, D. (2012). Ingeniería de software: Metodología XP. From <http://pnfiingenieriadesoftwaregrupocuatro.blogspot.com/2012/07/bienvenidos-al-blog.html>
- Managerial, Software Analyzer. (2005). Managerial Analyzer: Managerial Analyzer Software. From <http://www.managerialanalyzer.com/web/index.php>
- Menéndez, Yisel Arelis de la Cruz. (2010). *Definición espacial de redes hidráulicas de abasto en la actividad de proyecto de la EIPH Holguín "Raudal": Módulo de un Sistema CAD distribuido*. Oscar Lucero Moya, Holguín.
- MIC. (2011). Software Contables-Financieros: Agencia de Supervisión y Control. From http://www.acs-mic.cu/sw_certificados.htm
- Patón, M.y.F., Eduardo. (2002). Metodologías de Desarrollo software. From <http://alarcos.inf-cr.uclm.es/doc/ISOFTWAREI>
- Pereda, J. (2007). Consideraciones previas para diseñar una web. Retrieved 21-01-10, 2010, from <http://jimpereda.wordpress.com>
- Portellez, Liset Nápoles. (2010). *Módulo de Gerencia CAD de un sistema CAD distribuido para el diseño de redes hidráulicas de abasto en la EIPH Holguín "Raudal"*. . Oscar Lucero Moya, Holguín.
- Pressman., R. (1998). *Ingeniería del software. Un enfoque práctico* (Cuarta ed.). Madrid: McGraw-Hill.
- Quintero, Yisel Clavel. (2010). *Sistema de clasificación automática de noticias a publicar en el periódico ahora digital*.

. Oscar Lucero Moya, Holguín.

Rendón, A. (2004). *Desarrollo de sistemas informáticos usando UML y RUP*. Popayán, Colombia.

Día 1 “Python y sus ventajas” (2008).

Rodríguez C, T.M. (2008). *Sistema de Gestión de Información Editorial. Semanario Ahora*. (Trabajo de Diploma Optando por el título de Ingeniero Informático).

RODRÍGUEZ, H. (2012). Cuba avanza en la implementación del software libre.

Rosalw, R. (2007). Los diferentes lenguajes de programación para la web. Retrieved 22-02-10, 2010, from www.maestrosdelweb.com

Rosenberg, D., & Scott, K. (2001). *Applying Use Case Driven Object Modeling with UML* Addison Wesley.

Rosenberg, D. and M. Stephens. (2007). *Use Case Driven Object Modeling with UML. Theory and Practice*.

Toledo, Yanet Coya. (2010). *Sistema para el control de la producción científica de la Universidad de Holguín.*, Oscar Lucero Moya, Holguín.

Vegas, J.). Introducción a las Aplicaciones Web. Retrieved 21-01-10, 2010, from <http://www.infor.uva.es/jvegas/cursos/buendia/pordocente/node11.html>

Venzant, Manuel Barrera. (2010). *Sistema para la Gestión de Información de la Estrategia de Trabajo Metodológico en la UHOLM*. Oscar Lucero Moya, Holguín.

Zarete, S. (2008). Django el Framework Web Definitivo.

Zukowski, J. (2006). *Programación Java 2 J2SE 1.4*. (Vol. 1, 3). La Habana.: Editorial Félix Varela.

Anexos

Anexo 1. Descripción textual de los casos de uso

Caso de Uso: Iniciar Sesión
Curso Básico: El Usuario teclea su usuario y contraseña en la pantalla autenticación y da clic en el botón Entrar al sistema, el sistema valida que no existan campos vacíos o caracteres inadecuados y verifica que el usuario exista en la base de datos y que la contraseña sea la correcta mostrando la página principal de la aplicación.
Curso alterno: El usuario introduce los datos en un formato incorrecto, deja campos en blanco, introduce un usuario que no existente en la base de datos y una contraseña incorrecta. El sistema muestra un mensaje de error.

Tabla 5. Descripción textual de caso de uso Iniciar Sesión

Caso de Uso: Insertar Registro
Curso Básico: El especialista accede a la Interfaz principal del sistema da clic en el menú Registro de Cheques para visualizar todos los Cheques registrados, luego da clic en el botón Nuevo Cheque, el sistema muestra la Interfaz añadir Nuevo Cheque, el Especialista llena los campos con los datos correspondientes y da clic en el botón Crear Nuevo Cheque. El sistema verifica la validez de los datos a insertar. El Sistema Inserta los Datos.
Curso alterno: Campos vacíos: El Sistema Muestra el Mensaje de Notificación Campos Obligatorios y regresa a la Interfaz Añadir Cheque.

Tabla 6. Descripción textual de caso de uso Insertar Registro

Caso de Uso: Reporte Disponibilidad de Fondos
Curso Básico: El especialista accede al sistema da clic en Reportes, luego accede a los Reportes Generales y luego accede a el reporte de disponibilidad de Fondos.

Tabla 7. Descripción textual de caso de uso mostrar Disponibilidad de Fondos

Anexo 2. Diagramas de robustez

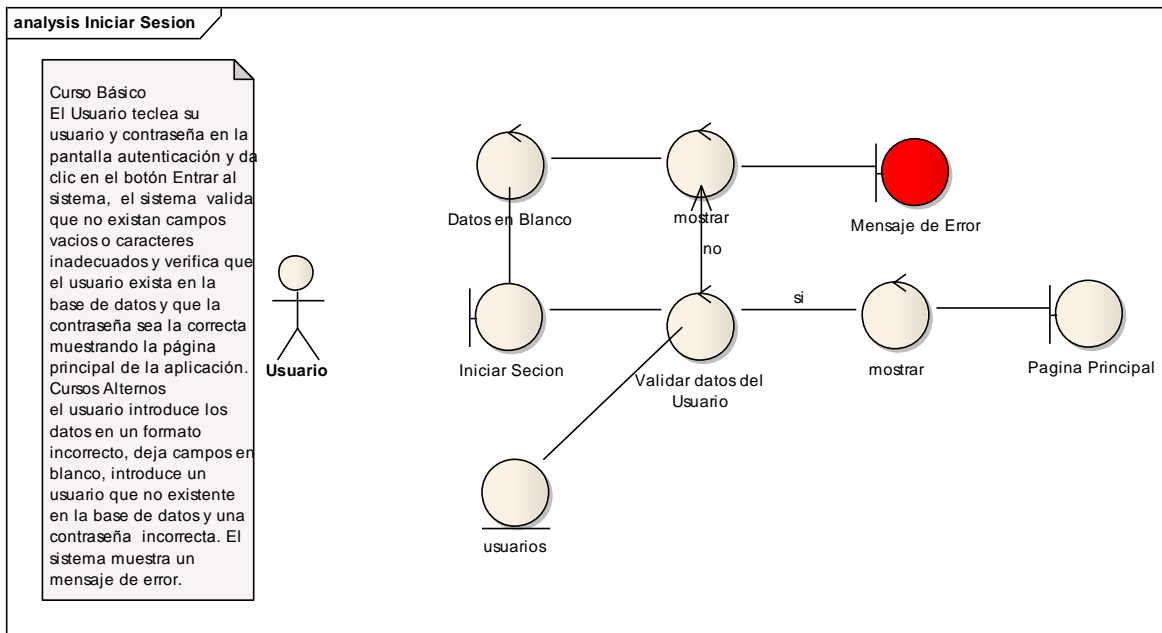


Figura 18. Iniciar Sesión

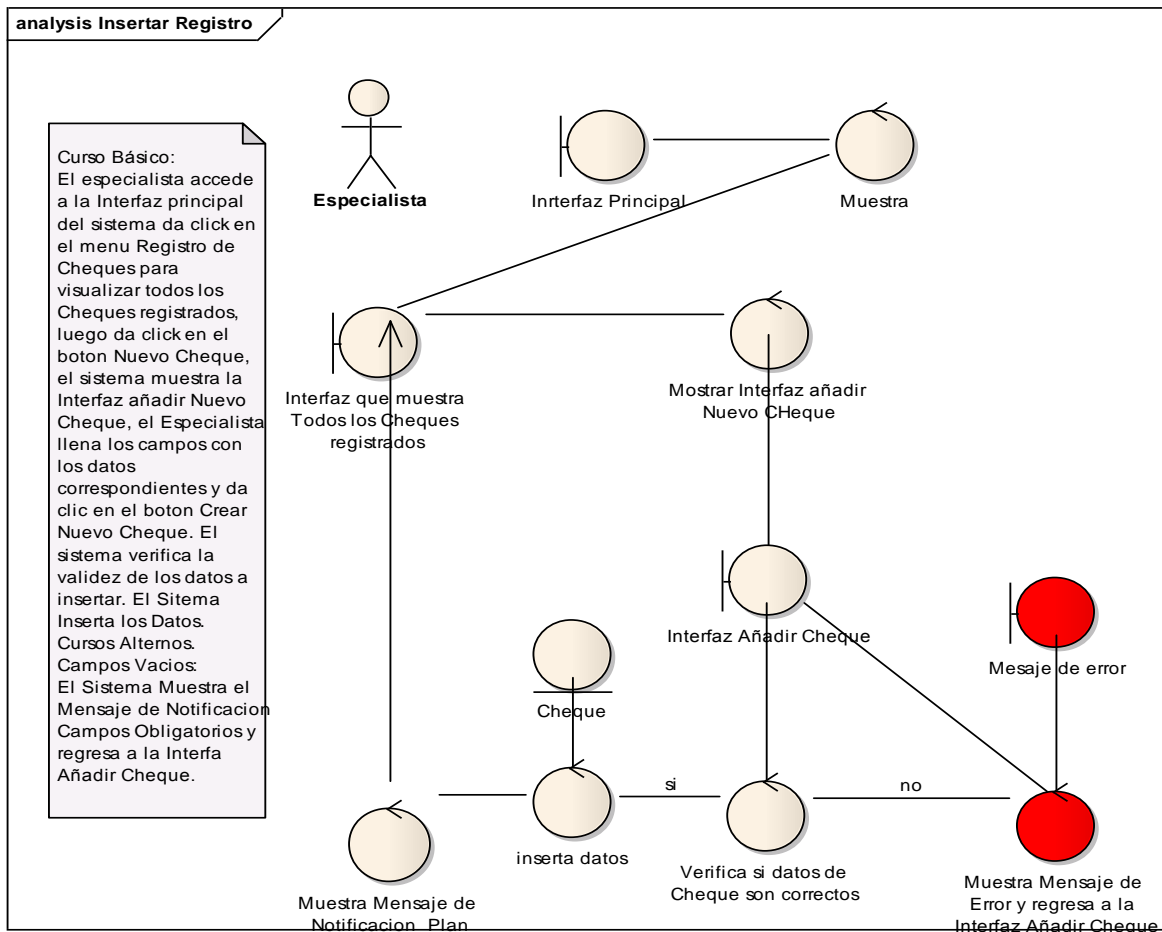


Figura 19. Insertar Registro

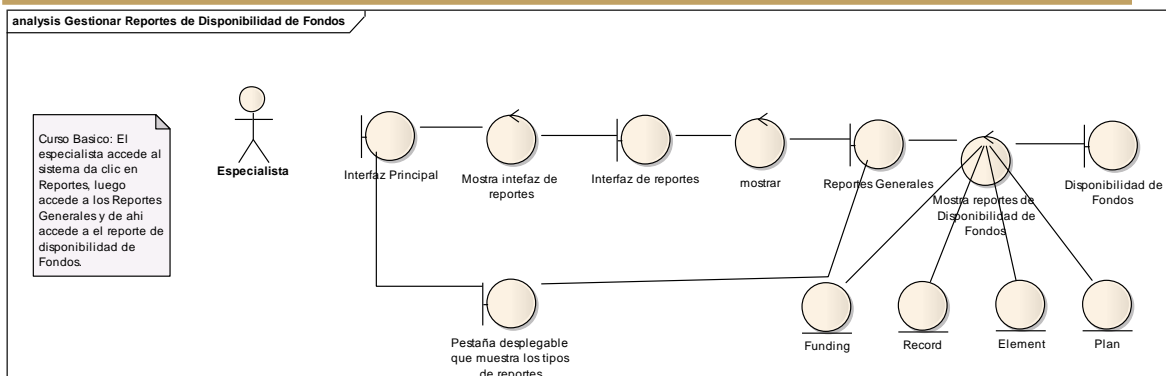


Figura 20. Reporte Disponibilidad de Fondos

Anexo 3. Diagramas de secuencia

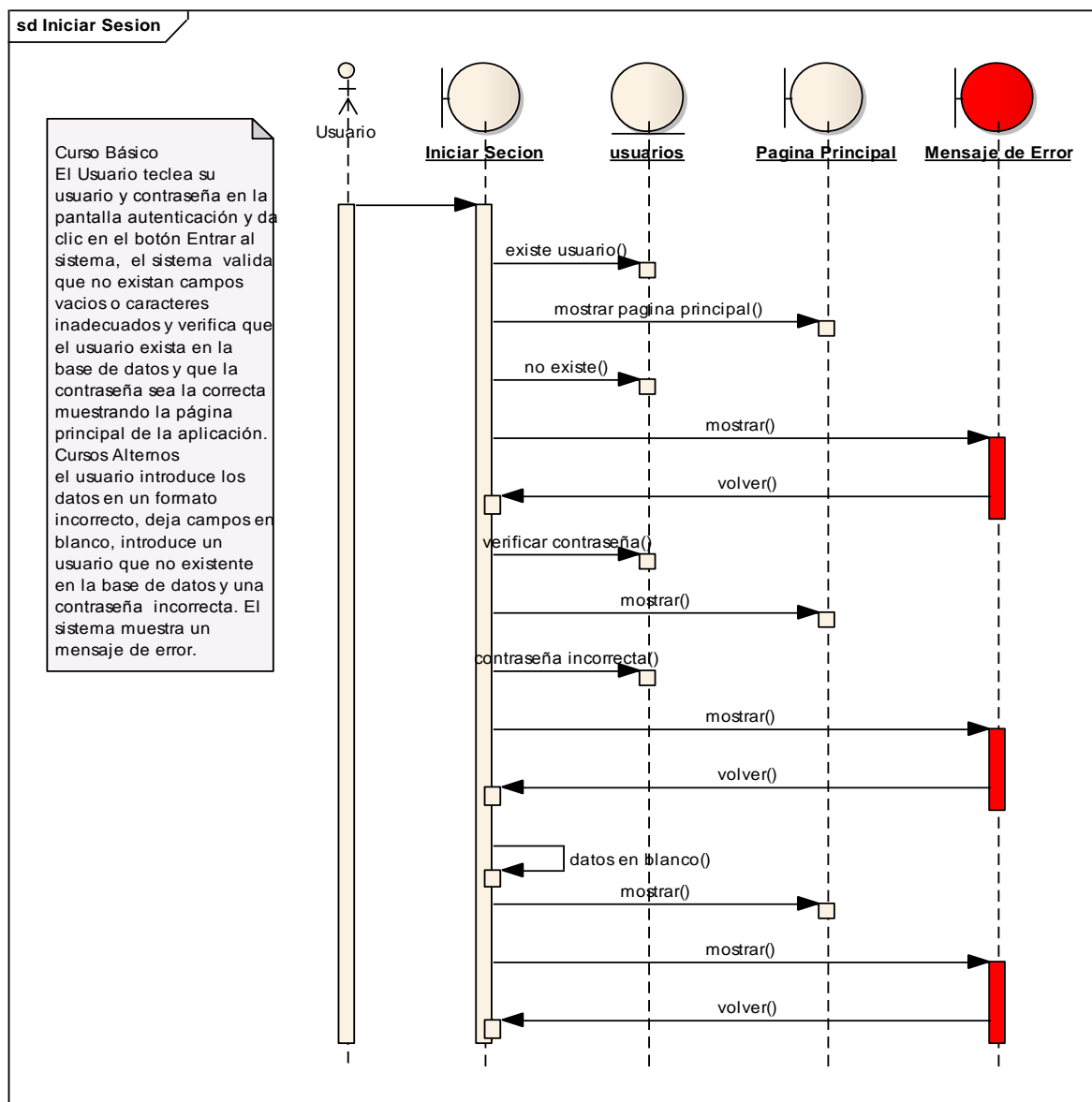


Figura 21. Iniciar Sesión

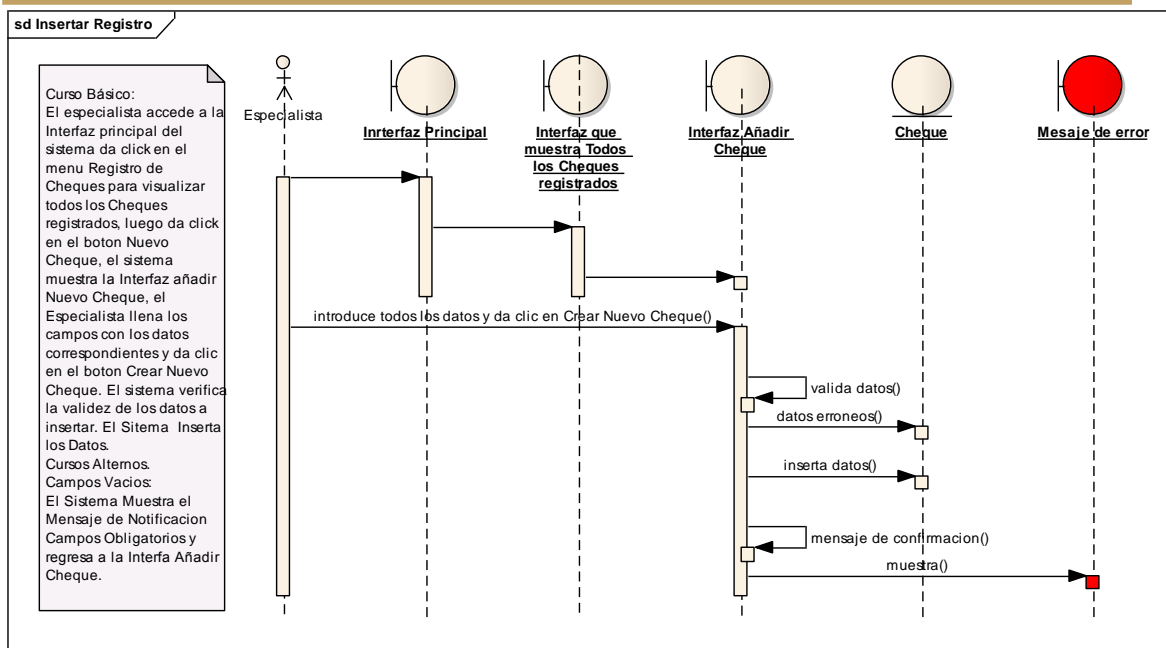


Figura 22. Insertar Registro

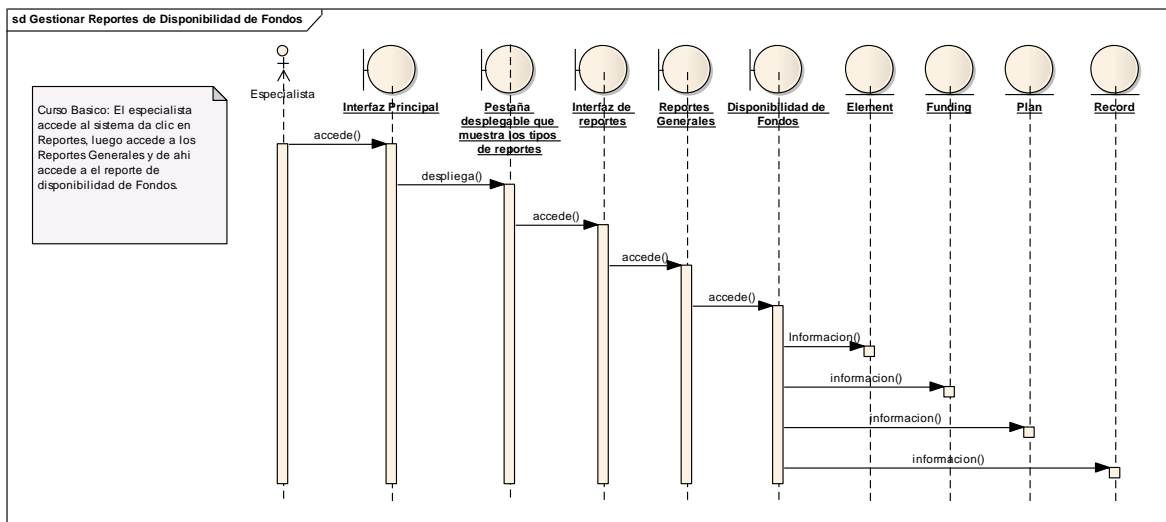


Figura 23. Reporte Disponibilidad de Fondos

Anexo 4. Diagrama de Modelo de Datos

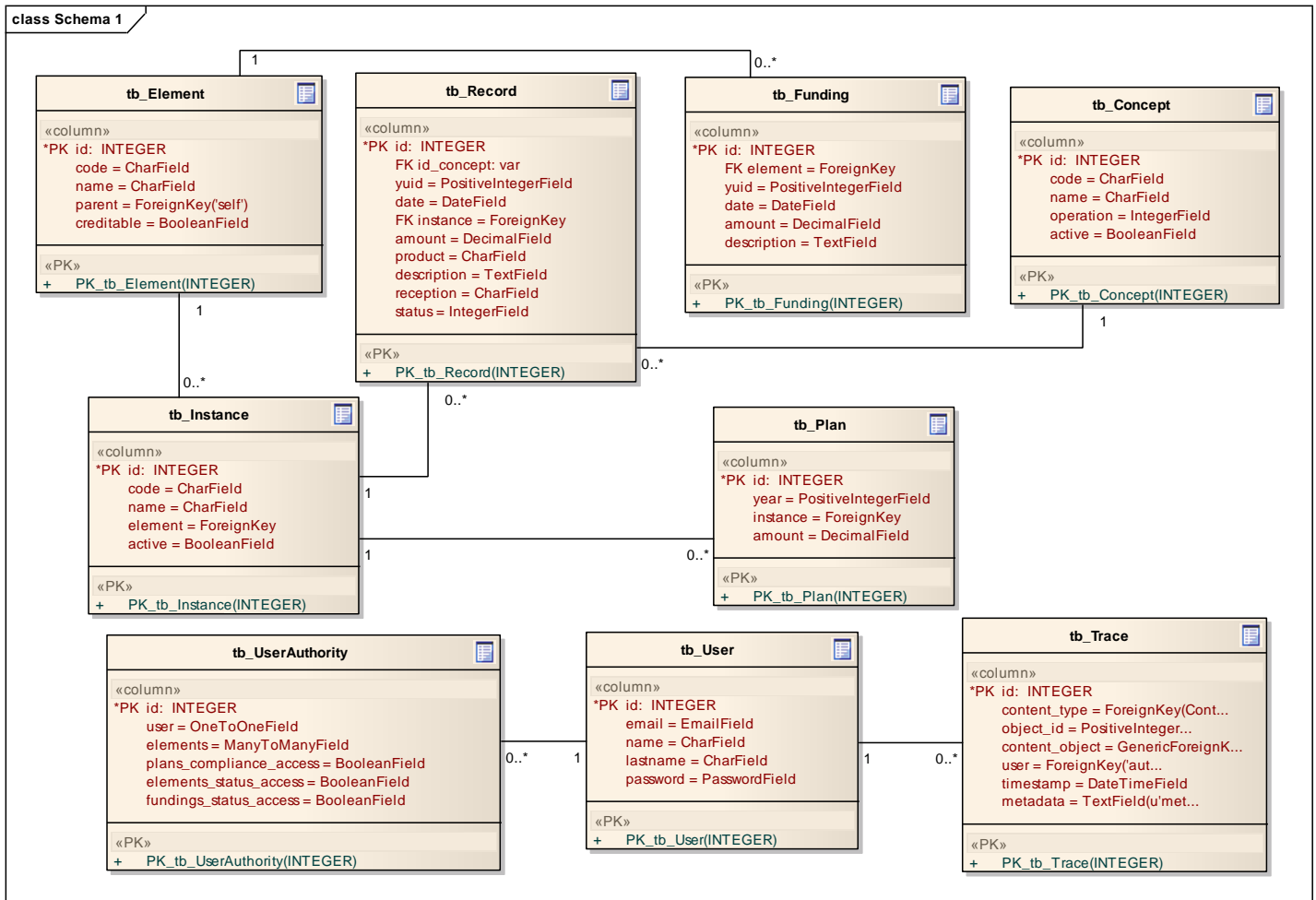


Figura 24. Modelo de Datos

Anexo 5. Análisis de puntos de casos de uso

Factor de complejidad técnica (TCF)

Este coeficiente se calcula mediante la cuantificación de un conjunto de factores que determinan la complejidad técnica del sistema. Cada uno de los factores se cuantifica con un valor de 0 a 5 de acuerdo a la opinión del equipo de desarrollo como se muestra en la tabla 8.

Metric	Description	Weight	Value	TCF
TCF01	Distributed System	2,00	0,00	0,00
TCF02	Response or throughput performance objectives	1,00	2,00	2,00
TCF03	End user efficiency (online)	1,00	2,00	2,00
TCF04	Complex internal processing	1,00	3,00	3,00
TCF05	Code must be re-usable	1,00	4,00	4,00
TCF06	Easy to install	0,50	1,00	0,50
TCF07	Easy to use	0,50	3,00	1,50
TCF08	Portable	2,00	0,00	0,00
TCF09	Easy to change	1,00	2,00	2,00
TCF10	Concurrent	1,00	2,00	2,00
TCF11	Includ special security features	1,00	3,00	3,00
TCF12	Provide direct access for third parties	1,00	5,00	5,00
TCF13	Special user training facilities are required	1,00	1,00	1,00
			Total:	26,00

Tabla 8. Factor de Complejidad Técnica

El Factor de complejidad técnica resulta:

Factor	Value
Unadjusted TCF value (UTV)	26,00
TCF Weighting (TWF)	0,01
TCF Constant (TC)	0,60
Technical Complexity Factor (TCF) = TC + (UTV * TWF)	0,86

Tabla 9. Valor del TCF

Factor de ambiente (EF)

Las habilidades y el entrenamiento del grupo involucrado en el desarrollo tienen un gran impacto en las estimaciones de tiempo. Estos factores son los que se contemplan en el cálculo del Factor de ambiente. El cálculo del mismo es similar al cálculo del Factor de complejidad técnica, es decir, se trata de un conjunto de factores que se cuantifican con valores de 0 a 5 como se muestra en la tabla 10.



Metric	Description	Weight	Value	TCF
ECF01	Familiar with Rational Unified Process	1,50	4,00	6,00
ECF02	Application experience	0,50	3,00	1,50
ECF03	Object-oriented experience	1,00	4,00	4,00
ECF04	Lead analyst capability	0,50	2,00	1,00
ECF05	Motivation	1,00	3,00	3,00
ECF06	Stable requirements	2,00	2,00	4,00
ECF07	Part-time workers	-1,00	0,00	-0,00
ECF08	Difficult programming language	-1,00	3,00	-3,00
			Total:	16,50

Tabla 10. Factor de Ambiente

El Factor de ambiente resulta:

Factor	Value
Unadjusted ECF value (UEV)	16,50
ECF Weighting (EWF)	-0,03
ECF Constant (EC)	1,40
Environmental Complexity Factor (ECF) = EC + (UEV * EWF)	0,90

Tabla 11. Valor del EF

Este análisis ha permitido conocer el costo final del software el cual se muestra en la tabla 12.

Item	Value
Date of Estimation	16-may-2014 21:18:51
Phase	*
Total Use Cases	51
Unique Use Case Points (UUCP)	255,00
Technical Complexity (TCF)	0,86
Environmental Complexity(ECF)	0,90
Use Case Points (UUCP * TCF * ECF) = UCP	198,00
Estimated Hours per UUCP (HRS)	6,00
Total Hours (HRS * UCP)	1188,00
Total Cost	4752,00

Tabla 12. Valor del Esfuerzo

Anexo 6. Pruebas realizadas al sistema

Buscar Plan

Caso de uso:	Buscar Plan.
Caso de prueba:	Encontrar planes que pertenecen a un indicador especificado
Entrada:	El especialista necesita la relación de planes pertenecientes al Indicador especificado. El indicador especificado no tiene planes asignados
Resultado:	Se muestra un mensaje al especialista informándole que “No hay planes que coincidan con su búsqueda”.

Tabla 13. *Caso de Prueba Buscar Plan*

Respuesta del sistema



Finanzas CUC Planes Registros de Cheques Asignaciones Reportes Ernesto

Planes

Nuevo plan inversiones Buscar Búsqueda avanzada

Estos resultados fueron filtrados. [Haga clic aquí para mostrar todos los elementos \(17\)](#)

Año	Indicador	Plan
No hay planes que coincidan con su búsqueda		

Figura 25. *Caso de prueba Buscar Plan*

Buscar Plan

Caso de uso:	Buscar Plan (Búsqueda Avanzada).
Caso de prueba:	Encontrar planes que pertenecen a un año especificado
Entrada:	El especialista necesita la relación de planes pertenecientes al año especificado. El año especificado no tiene planes asignados
Resultado:	Se muestra un mensaje al especialista informándole que “No hay planes que coincidan con su búsqueda”.

Tabla 14. Caso de Prueba Buscar Plan

Respuesta del sistema

Finanzas CUC
 Planes
 Registros de Cheques
 Asignaciones
 Reportes
 Ernesto

Planes

Nuevo plan

Buscar
Búsqueda avanzada

Estos resultados fueron filtrados. [Haga clic aquí para mostrar todos los elementos \(17\)](#)

Año	Indicador	Plan
No hay planes que coincidan con su búsqueda		

Figura 26. Caso de prueba Buscar Avanzada de Planes

Insertar Asignación

Caso de uso:	Insertar Asignación.
Caso de prueba:	Insertar en el sistema una nueva Asignación con un campo con llenado incorrecto.
Entrada:	El especialista necesita insertar en el sistema una nueva asignación. La asignación tiene un llenado incorrectamente.
Resultado:	Se muestra un mensaje al especialista: “Introduzca un número”

Tabla 15. Caso de Prueba Insertar asignación

Respuesta del sistema

Finanzas CUC
Planes
Registros de Cheques
Asignaciones
Reportes
Ernesto

Nueva asignación

Fecha:

Fecha de emisión

Elemento de gasto:

Elemento de gasto al que pertenece esta asignación

Importe:

- Introduzca un número.

Importe de la entrada

Descripción:

Descripción de la entrada

Figura 27. Caso de prueba Nueva Asignación

Listar Cheques Registrados

Caso de uso:	Listar Cheques Registrados.
Caso de prueba:	Listar todos los cheques registrados.
Entrada:	El especialista desea visualizar la lista de Cheques Registrados. El sistema no contiene Cheques Registrados.
Resultado:	Se muestra un mensaje al especialista informándole que “No hay registros”.

Tabla 16. Caso de Prueba Listar Cheques Registrados

Respuesta del sistema

Finanzas CUC
Planes
Registros de Cheques
Asignaciones
Reportes
Ernesto

Cheques

Nuevo cheque

Buscar
Búsqueda avanzada

Registro	Fecha	Suministrador	Producto	Clasificación (Indicador)	Concepto	Importe	Inf. Recepción	Est.
2-2014	30-05-2014	ECO A 19	Marco de aluminio para ventanas	Mantenimiento	Reparacion de la Universidad	400.00	cvt	✓
1-2014	30-05-2014	Copextel	monitores	Otros Servicios de Residentes	Reparacion de Computadoras	400.00	servicio	✓

Figura 28. Caso de prueba Listar Cheques

Anexo 7. Encuesta para la determinación de expertos.

Nombre y apellidos: _____ Fecha: _____

Usted ha sido seleccionado como posible experto para ser consultado respecto al grado de relevancia de las posibilidades que brinda la aplicación Web **Sistema de apoyo a la gestión de la información del Plan de la Economía de la Universidad de Holguín “Oscar Lucero Moya”**. Se necesita determinar su coeficiente de competencia en el tema, a los efectos de reforzar la validez del resultado de la consulta que se realizará. Por esta razón se le solicita que responda las siguientes preguntas de la forma más objetiva que le sea posible.

1. Marque con una cruz (X), en la tabla siguiente, el valor que se corresponde con el grado de conocimientos que usted posee sobre el tema. Considere que la escala que se presenta es ascendente, es decir, el conocimiento sobre el tema referido va creciendo desde 0 hasta 10.

0	1	2	3	4	5	6	7	8	9	10

2. Realice una autovaloración del grado de influencia de cada una de las fuentes que se muestran a continuación, ha tenido en su conocimiento y criterio sobre los indicadores que se han puesto a su consideración. Para ello marque con una cruz (X), según corresponda, en **A** (Alto), **M** (Medio) o **B** (bajo).

Fuentes de Argumentación	Grado de influencia de cada una de las fuentes		
	A	M	B
Análisis teóricos realizados por usted			
Su experiencia obtenida			
Trabajo de autores nacionales			
Trabajo de autores extranjeros			
Su propio conocimiento del estado del problema en el extranjero			
Su intuición			

Gracias por su cooperación.

Anexo 8. Encuesta aplicada a los expertos para la valoración del sistema.

Estimado compañero(a), esta encuesta se aplica con el propósito de recopilar opiniones y sugerencias acerca de la aplicación Web **Sistema de apoyo a la gestión de la información del Plan de la Economía de la Universidad de Holguín “Oscar Lucero Moya”**. Le pedimos por favor que lea cuidadosamente las preguntas siguientes, y exprese su criterio. Su opinión es fundamental.

Fecha: _____.

1. A continuación sometemos a su valoración una serie de elementos que se consideran importantes en la evaluación del sistema.

Para esta evaluación se le propone las siguientes categorías:

- Muy Adecuado (**MA**)
- Bastante Adecuado (**BA**)
- Adecuado (**A**)
- Poco Adecuado (**PA**)
- No Adecuado (**NA**)

Marque con una **X** la categoría que considera adecuada para cada criterio:

Criterios	MA	BA	A	PA	NA
¿La aplicación Web favorece la gestión de la información del Plan de la Economía de la Universidad de Holguín “Oscar Lucero Moya”?					
¿Cómo valora la eficiencia del sistema con respecto a los procesos que se realizaban de forma manual anteriormente?					
¿Qué le parece el uso de los colores e imágenes del sistema?					
¿Qué tan “amigable” le resultó el diseño de las interfaces del sistema?					
¿Cómo considera usted el acceso a las funcionalidades del sistema?					
¿Cómo valora el uso que se le puede dar al sistema?					

2. ¿Qué otros elementos novedosos encuentra en la aplicación?

3. Tiene alguna otra idea o sugerencia que no se haya abordado en esta encuesta.

Gracias por su cooperación.

Anexo 9. Procesamiento de la encuesta de opinión de los usuarios finales aplicándose el método Delphi.

Tabla de Frecuencia Absoluta						
Criterios	MA	BA	A	PA	NA	TOTAL
¿La aplicación Web favorece la gestión de la información del Plan de la Economía de la Universidad de Holguín “Oscar Lucero Moya”?	5	1	0	0	0	6
¿Cómo valora la eficiencia del sistema con respecto a los procesos que se realizaban de forma manual anteriormente?	4	2	0	0	0	6
¿Qué le parece el uso de los colores e imágenes del sistema?	3	3	0	0	0	6
¿Qué tan “amigable” le resultó el diseño de las interfaces del sistema?	5	1	0	0	0	6
¿Cómo considera usted el acceso a las funcionalidades del sistema?	4	2	0	0	0	6
¿Cómo valora el uso que se le puede dar al sistema?	5	1	0	0	0	6



Tabla de Frecuencia Absoluta Acumulada					
Criterios	MA	BA	A	PA	NA
¿La aplicación Web favorece la gestión de la información del Plan de la Economía de la Universidad de Holguín “Oscar Lucero Moya”?	5	6	6	6	6
¿Cómo valora la eficiencia del sistema con respecto a los procesos que se realizaban de forma manual anteriormente?	4	6	6	6	6
¿Qué le parece el uso de los colores e imágenes del sistema?	3	6	6	6	6
¿Qué tan “amigable” le resultó el diseño de las interfaces del sistema?	5	6	6	6	6
¿Cómo considera usted el acceso a las funcionalidades del sistema?	4	6	6	6	6
¿Cómo valora el uso que se le puede dar al sistema?	5	6	6	6	6

Tabla del Inverso de la Frecuencia Absoluta Acumulada				
Criterios	MA	BA	A	PA
¿La aplicación Web favorece la gestión de la información del Plan de la Economía de la Universidad de Holguín “Oscar Lucero Moya”?	0,8333	1	1	1
¿Cómo valora la eficiencia del sistema con respecto a los procesos que se realizaban de forma manual anteriormente?	0,6667	1	1	1
¿Qué le parece el uso de los colores e imágenes del sistema?	0,5	1	1	1
¿Qué tan “amigable” le resultó el diseño de las interfaces del sistema?	0,8333	1	1	1
¿Cómo considera usted el acceso a las funcionalidades del sistema?	0,6667	1	1	1
¿Cómo valora el uso que se le puede dar al sistema?	0,8333	1	1	1

Tabla de Determinación de los Puntos de Corte
--



Criterios	MA	BA	A	PA	Suma	Promedio	N - Prom.
¿La aplicación Web favorece la gestión de la información del Plan de la Economía de la Universidad de Holguín "Oscar Lucero Moya"?	0,97	3,49	3,49	3,49	11,44	2,86	-0,09
¿Cómo valora la eficiencia del sistema con respecto a los procesos que se realizaban de forma manual anteriormente?	0,43	3,49	3,49	3,49	10,9	2,73	0,04
¿Qué le parece el uso de los colores e imágenes del sistema?	0	3,49	3,49	3,49	10,47	2,62	0,15
¿Qué tan "amigable" le resultó el diseño de las interfaces del sistema?	0,97	3,49	3,49	3,49	11,44	2,86	-0,09
¿Cómo considera usted el acceso a las funcionalidades del sistema?	0,43	3,49	3,49	3,49	10,9	2,73	0,04
¿Cómo valora el uso que se le puede dar al sistema?	0,97	3,49	3,49	3,49	11,44	2,86	-0,09
Suma	3,77	20,94	20,94	20,94	66,59		
Punto de corte	0,63	3,49	3,49	3,49	11,1	2,77	= N (Prom. Gen.)

Conclusiones Generales					
Criterios	MA	BA	A	PA	NA
¿La aplicación Web favorece la gestión de la información del Plan de la Economía de la Universidad de Holguín "Oscar Lucero Moya"?	Si	-	-	-	-
¿Cómo valora la eficiencia del sistema con respecto a los procesos que se realizaban de forma manual anteriormente?	Si	-	-	-	-
¿Qué le parece el uso de los colores e imágenes del sistema?	Si	-	-	-	-
¿Qué tan "amigable" le resultó el diseño de las interfaces del sistema?	Si	-	-	-	-
¿Cómo considera usted el acceso a las funcionalidades del sistema?	Si	-	-	-	-
¿Cómo valora el uso que se le puede dar al sistema?	Si	-	-	-	-