

UNIVERSIDAD NACIONAL DE INGENIERÍA
RECINTO UNIVERSITARIO PEDRO ARAUZ PALACIOS

Mon
025.04
B732
2009



FACULTAD DE CIENCIAS Y SISTEMAS

INFORME DE RESULTADOS FINALES PRÁCTICA PROFESIONAL

“ANÁLISIS, DISEÑO Y DESARROLLO DEL SISTEMA AUTOMATIZADO DE CONTROL DE BIENES, INETER”

Presentado por: Br. Rubenia Olaska Borge Flores

Tutor: Ing. Evelyn Espinoza Aragón

Managua, 31 de marzo de 2009

Índice

<i>Contenido</i>	<i>Página</i>
I. Introducción	1
II. Descripción del Trabajo	2
1. Objetivos	2
2. Alcances Técnicos	3
3. Aspectos Tecnológicos	9
4. Cronograma de Actividades	10
5. Análisis de Costos. Impacto Técnico-económico y social.	11
III. Conclusiones	25
IV. Anexos	26
V. Bibliografía	326



I. INTRODUCCIÓN

La automatización de sistemas es una tarea de vital importancia para lograr el cumplimiento de las metas de una organización. Dentro de una organización con tanta envergadura como lo es el Instituto Nicaragüense de Estudios Territoriales, los beneficios de la automatización en conjunto con el aprovechamiento de Tecnologías de la Información pueden brindar ventajas competitivas como son:

- Reducción en los costos en los productos y servicios ofrecidos.
- Incorporación de nuevas características que permitan diferenciar los productos y aumentar su valor agregado.
- Innovación en la creación de nuevos productos que abran las puertas a nuevos mercados, promoviendo así el crecimiento, la expansión y la diversificación.
- Desarrollo de alianzas estratégicas con otras organizaciones a través de Sistemas de Información basados en Internet.

Los proyectos de Automatización de Sistemas deben estar acompañados de un buen plan de trabajo en las áreas de Análisis, Diseño y Desarrollo para obtener como resultado productos software que satisfagan las necesidades de la empresa.

El éxito de un proyecto de Desarrollo de Software está en dependencia de la calidad en el Análisis y Diseño más que en la Programación. En el ciclo de vida de desarrollo de un sistema, el 80% de las actividades están destinadas al Análisis y Diseño, y solamente se dedica un 20% a la Programación.

La Práctica Profesional “Análisis, Diseño y Desarrollo del Sistema Automatizado de Control de Bienes en INETER” tiene su alcance técnico compuesto por:

- El desarrollo de un proceso investigativo con el objetivo de estudiar la problemática y la necesidad de automatización de sistemas en el Departamento de Control de Bienes.
- La aplicación de una Metodología de Software Orientada a Objetos basada en UML, desarrollada por el señor Grady Booch donde se utilizan como herramientas los diagramas del Lenguaje Unificado de Modelado en las etapas de Análisis, Diseño, Desarrollo e Implementación de un Proyecto de Software. Cabe mencionar que el alcance de este trabajo abarca solamente las etapas de Análisis, Diseño y Desarrollo.
- La utilización del Diagrama Entidad-Relación y el Proceso de Normalización para elaborar el Diseño de la Base de Datos Relacional del Sistema Automatizado.



II. DESCRIPCIÓN DEL TRABAJO

Objetivos de la Práctica Profesional

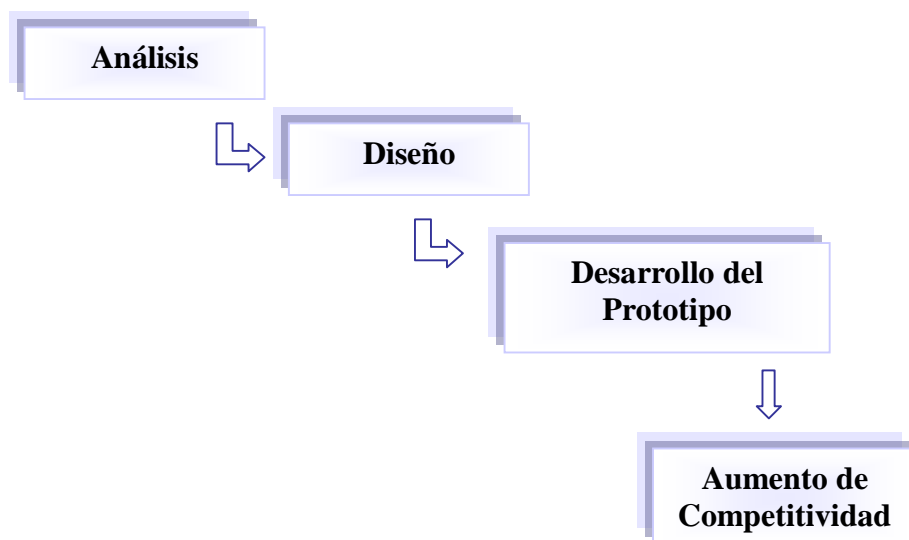
Objetivo General

Realizar el Análisis, Diseño y Desarrollo del Prototipo del Sistema Automatizado para el Control de Activos Fijos en el Instituto Nicaragüense de Estudios Territoriales -*INETER* que sirva como una herramienta para agilizar las transacciones, aumentar la confiabilidad, seguridad en el almacenamiento y recuperación de la información, asimismo que reducir la cantidad de errores en el procesamiento de los datos, que ocurren en las actividades diarias del Departamento de Control de Bienes. También, desarrollar un prototipo de sistema que contribuya en la satisfacción de la demanda de información para la toma de decisiones en el instituto.

Objetivos Específicos

- Realizar un Análisis de Sistemas en el Departamento de Control de Bienes.
- Construir un Diseño Detallado del Sistema.
- Desarrollar el Prototipo del Sistema Automatizado.

Esquema de Objetivos





ALCANCES TÉCNICOS

El alcance técnico de la Práctica Profesional abarca el desarrollo de un proyecto de investigación científica con el objetivo de conocer y analizar la problemática presentada en el Departamento de Control de Bienes del Instituto Nicaragüense de Estudios Territoriales, *INETER* y la utilización de la Metodología de Desarrollo de Software Orientada a Objetos basada en UML, propuesta por Gary Booch en las etapas de Análisis, Diseño y Desarrollo del Prototipo en el Sistema Automatizado de Control de Bienes. Además se utilizan el Diagrama de Entidad-Relación y el Proceso de Normalización para realizar el diseño de la Base de Datos Relacional. Aunque estas herramientas no son propias de la Metodología, son muy útiles y pueden aportar mucho para el diseño de la Base de Datos.

El desarrollo de la investigación científica consiste en el uso del método científico para obtener información importante y objetiva sobre la problemática que enfrenta el Departamento de Control de Bienes.

La Metodología de Software Orientada a Objetos basada en UML consiste en cuatro etapas principales, las cuales son:

- Análisis de Requerimientos
- Diseño del Sistema
- Diseño Detallado
- Implementación y Pruebas



A continuación se Presenta la Metodología con sus etapas y los diagramas de UML utilizados en cada una de ellas.

Metodología basada en UML

Análisis de Requerimientos

- Descubrimiento de los procesos de negocio
- Descripción de los procesos de negocio
- Diagramas de actividades
- Diagrama de clases de alto nivel
- Diagramas de casos de uso de alto nivel

Diseño del Sistema

- Diagramas de casos de uso depurados
- Plantillas de colleman
- Diagramas de secuencia y diagramas de colaboración
- Diagramas de paquete
- Diagramas de estado

Diseño Detallado

- Diagramas de Componentes
- Diagrama de Despliegue

Implementación y Pruebas

- Elaboración del prototipo
- Definir estándares de programación
- Codificación y Pruebas Unitarias
- Pruebas de Módulos y de Sistemas

En la etapa de Implementación y Pruebas el alcance técnico de esta práctica profesional abarca solamente hasta la fase de Elaboración del Prototipo debido a razones financieras y limitaciones de tiempo.



Análisis de Requerimientos

El análisis de requerimientos es la primera etapa de la metodología de desarrollo de software. En esta etapa se descompone el sistema en sus piezas o componentes con el propósito de estudiar la forma en que estas partes que componen el sistema, trabajan e interactúan para lograr su propósito principal.

Para realizar este análisis utilizamos las herramientas siguientes:

- Descubrimiento de los procesos de negocio
- Descripción de los procesos de negocio
- Diagramas de actividades
- Diagrama de clases de alto nivel
- Diagramas de casos de uso de alto nivel

El descubrimiento de los procesos de negocios se logra mediante la elaboración de entrevistas a los empleados con el objetivo de comprender todos los procesos que se realizan en el Departamento de Control de Bienes y cuales son sus pasos uno a uno.

Una vez que los procesos se han descubierto, se procede a describirlos y a realizar un diagrama de actividades por cada uno de los procesos encontrados. En los diagramas de actividades se describe gráficamente el flujo secuencial de los procesos de negocios.

El diagrama de clases de alto nivel y todas las asociaciones existentes entre las clases sirven para analizar y comprender el dominio del sistema. El diagrama de clases es muy útil puesto que describe la estructura de los objetos del sistema y muestra las clases de objetos que componen al sistema, así como las relaciones existentes entre estos objetos.

Finalmente, en esta etapa se realizan los casos de uso de alto nivel en los que se especifica cuál es la funcionalidad actual del sistema.

Diseño del Sistema

En el Diseño del Sistema se utilizan técnicas de resolución de problemas complementarias al Análisis de Sistemas para separar los componentes del sistema y luego formar nuevamente un sistema completo –que es un sistema esperado y mejorado. Este nuevo sistema puede incluir adiciones, eliminaciones y cambios en relación con el sistema original del Departamento de Control de Bienes.



Las herramientas utilizadas en el Diseño del Sistema son:

- Diagramas de casos de uso depurados
- Plantillas de colleman
- Diagramas de secuencia y diagramas de colaboración
- Diagramas de paquete
- Diagramas de Estado

Con los diagramas de caso de uso se muestra gráficamente la forma en que los empleados del Departamento de Control de Bienes interactúan con el sistema así como también la interacción con los usuarios indirectos y sistemas externos.

Seguidamente se elabora una plantilla de Coleman para cada caso de uso presentado donde se detalla textualmente la secuencia de pasos de cada interacción del caso de uso.

Con los diagramas de secuencia se describe la forma en que los objetos interactúan unos con otros a través del envío de mensajes en la ejecución de los casos de uso y luego se utilizan los diagramas de colaboración con los que se encuentra la interacción de los objetos dentro de una red.

Con los diagramas de paquete se organizan los elementos de diagramas de casos de uso en diferentes grupos o subsistemas.

La última actividad dentro del Diseño del Sistema es la elaboración de los Diagramas de Estado en los que se muestra el ciclo de vida de cada objeto, es decir todos los estados en los que se puede encontrar un objeto y los eventos que pueden causar que el objeto pase de un estado a otro.



Diseño Detallado

El diseño Detallado muestra las especificaciones físicas de una solución de software.

En esta etapa se concretan los detalles en cuanto a Software y Hardware del producto software que satisface las necesidades de información del Departamento de Control de Bienes. Para esto se utilizan los siguientes diagramas:

- Diagramas de Distribución o Diagramas de Despliegue
- Diagramas de Componentes

Los diagramas de Distribución sirven para describir la arquitectura física del hardware y software del sistema. Describen los componentes de software, procesadores y dispositivos que forman parte de la arquitectura del Sistema Automatizado de Control de Bienes.

Posteriormente, para describir gráficamente la arquitectura física del sistema se utilizan los Diagramas de Componentes. También se usan los Diagramas de Componentes para mostrar la forma en que el código de programación se divide en módulos o componentes.

Implementación y Pruebas

- Elaboración del prototipo
- Definir estándares de programación
- Codificación y Pruebas Unitarias
- Pruebas de Módulos y de Sistemas

Esta es la última etapa de la Metodología de Desarrollo de Software Orientada a Objetos basada en UML. El alcance de la Práctica Profesional Análisis, Diseño y Desarrollo del Sistema Automatizado de Control de Bienes en INETER contempla hasta la elaboración del prototipo debido a limitaciones de tiempo y recursos.

En el desarrollo del prototipo se diseñan las interfaces gráficas de usuario para el sistema y se utiliza una herramienta para desarrollar aplicaciones que acceden a los datos almacenados en la base de datos.



Diseño de la Base de Datos

Para realizar el diseño de la base de datos se utiliza el Modelo Entidad-Relación y el Proceso de Normalización.

El modelo Entidad-Relación es útil para modelar una percepción del mundo real sobre el límite del dominio estudiado, en este caso, el Departamento de Control de Bienes. La percepción del mundo real obtenida se compone de un conjunto de objetos básicos llamados entidades y de relaciones entre estos objetos.

Para analizar los atributos de los datos y agruparlos de forma no redundante, estable, flexible y adaptable se recurre al Proceso de Normalización.

El proceso de normalización se lleva hasta la tercera forma normal. En la primera forma normal se eliminan los grupos repetidos y se identifica la llave primaria; en la segunda se asegura que todos los atributos que no son llave sean completamente dependientes de la llave primaria. En la tercera forma normal se elimina cualquier dependencia transitiva, -una dependencia transitiva es aquella en la cual atributos que no son llave son dependientes de otros atributos que no son llave.



ASPECTOS TECNOLÓGICOS

Todo sistema de computadoras consta de dos componentes principales que son Hardware y Software. El Hardware se refiere a todos los dispositivos físicos y electrónicos que conforman la computadora. Por otro lado, el Software se refiere a todos los programas que se instalan en la computadora para que esta pueda ser de utilidad.

Los Aspectos Tecnológicos en cuanto a Software que se deben ser utilizados en el análisis, diseño y desarrollo del Sistema Automatizado de Control de Bienes son los siguientes.

Herramienta Software	Utilidad
Microsoft Visual Studio	Programa para desarrollar aplicaciones para Microsoft Windows
Microsoft SQL Server	Gestor de Base de Datos
InstallShield	Aplicación para desarrollar Instaladores para Microsoft Windows
Microsoft Windows XP	Sistema Operativo
Microsoft Office 2007	Aplicaciones para Microsoft Windows
Crystal Reports 11	Diseñador de Reportes
Microsoft Visio	Software de Diagramación para Microsoft Windows
Microsoft Project	Software para Administración de Proyectos

Con respecto a los componentes Hardware que son necesarios:

Hardware	Función
Servidor de Bases de Datos	Almacenar la base de datos del Departamento de Control de Bienes
Computadoras Personales	Procesar los datos con usando el Sistema Automatizado de Control de Bienes
Impresora	Imprimir los reportes necesarios
Tarjetas de Red	Conectar las computadoras personales en red con el servidor de bases de datos



CRONOGRAMA DE ACTIVIDADES

Actividad	Tiempo	Total
Diseño de la Investigación Científica		
Elaboración del Prototipo	4	
Desarrollo de la Investigación	4	
Preparación del Informe Final de Resultados	4	12
Metodología basada en UML		
Análisis de Requerimientos		
Descubrimiento de los Procesos del Negocio	4	
Descripción de los Procesos del Negocio	4	
Diagramas de Actividades	2	
Diagramas de Clases de Alto Nivel	2	12
Diseño del Sistema		
Diagramas de Casos de Uso Depurados	1	
Plantillas de Colleman	8	
Diagramas de Secuencia y Diagramas de Colaboración	8	
Diagramas de Paquete	1	
Diagramas de Estado	2	20
Diseño Detallado		
Diagramas de Componentes	5	
Diagrama de Despliegue	1	6
Implementación y Pruebas		
Elaboración del Prototipo	12	
Definir Estándares de Programación		
Codificación y Pruebas Unitarias		
Pruebas de Módulos y de Sistemas		12
Diseño de la Base de Datos		
Elaboración del Modelo Entidad-Relación	2	
Normalización	2	4
Preparación del Informe Final de las Prácticas Profesionales		8
Total		66

Nota: Los tiempos están estimados en semanas.



ANÁLISIS DE COSTOS. IMPACTO TÉCNICO-ECONÓMICO Y SOCIAL

Análisis de Costos

En la actualidad son muchos los métodos existentes para la predicción o estimación de costos en proyectos de desarrollo de software. Sin embargo, estas estimaciones distan mucho de ser medidas exactas ya que el esfuerzo de desarrollar un software es siempre una actividad de predicción.

Se habla de predicción de costos porque lo que hacemos es estimar el tamaño de un software que no existe en el momento de la predicción.

Dentro de los modelos de estimación de costos para proyectos software se encuentran los siguientes:

- Estimación por COCOMO
- Estimación Wideband-Delphi
- Estimación con Lógica Difusa
- Estimación a través de Componentes Estándares
- Estimación a través del Factor de Complejidad
- Estimación PROBE
- Estimación a través de Puntos de Función

A continuación se describe brevemente las generalidades de cada uno de estos métodos:

Estimación por COCOMO:

El Modelo COCOMO siempre relaciona el esfuerzo E (meses-hombre) con el tamaño S (MLOCS) de acuerdo a: $E = a * S^b$ donde a y b son parámetros determinados por el tipo de software a ser desarrollado. Para usar este modelo para predecir el esfuerzo en la etapa de captura de requisitos, necesitamos primero determinar (predecir) los parámetros y luego el tamaño eventual del sistema.



Estimación Wideband-Delphi:

La idea fundamental de este modelo es usar varios expertos que hacen estimaciones independientes y luego hacerlos converger hacia una estimación única. El procedimiento es el siguiente: Primeramente, cada experto recibe las especificaciones del programa y un formulario de estimación. Seguidamente, se reúnen a conversar sobre suposiciones, dudas, etc., luego cada uno lista las tareas y produce una estimación y las estimaciones son recogidas por un moderador quien tabula los resultados y los devuelve a los expertos (estimaciones, promedio, mediana, etc.)

Finalmente, los expertos se reúnen nuevamente y discuten las tareas y se vuelve a hacer una nueva estimación. Es sabido que el método produce estimaciones bastante precisas debido a que la discusión entre los expertos a menudo clarifica aspectos y produce cambios en las estimaciones para brindar exactitud.

Estimación con Lógica Difusa:

Se basa en comparar con información histórica de productos anteriores. Se construye una tabla con rangos y sub-rangos de tamaño de proyectos previos. Se ordenan los proyectos según su tamaño usando una medida como LOC o FPs. Como se muestra en el ejemplo:

Tamaño	Complejidad		
	Bajo	Intermedio	Alto
Muy Pequeño	104	173	288
Pequeño	288	481	802
Mediano	802	1338	2230
Grande	2230	3719	6202
Muy Grande	6202	10341	1724316

Las filas indican el tamaño del proyecto (en LOC o PFs) y las columnas la complejidad de los programas o sistemas. El estimador selecciona una de las categorías y subcategorías comparando el nuevo proyecto con los proyectos anteriores dentro del rango considerado, luego toma como referencia principalmente aquellos proyectos previos que están en esa categoría.



Estimación a través de Componentes Estándares:

El método de Componentes Estándares se basa en mantener una base de datos histórica con información de componentes usados en proyectos previos, en varios niveles de abstracción: subsistemas completos, módulos, interfaces de usuario, etc.

Utilizando la información almacenada en la base de datos histórica, se estima cuántas de cada una de ellas habrá en el nuevo proyecto (estimado, máximo y mínimos). Después se combina esto, ponderando 4 veces el más probable y una vez los máximos y mínimos $(4*est+máx+min)/6$.

Componente	LOC	min	est	max	prob	LOC
Módulo A	932	11	18	22	17,5	16310
Módulo B	543	35	40	44	39,8	21611

Total						546359

Estimación PROBE:

Este método es aplicable en ambientes de OOP en las etapas de diseño, diseño conceptual, identificación de principales objetos (tipo, tamaño), calcular la proyección de LOC, estimar tamaño, calcular intervalo de predicción, etc.

A continuación se presenta un ejemplo aplicado en la etapa de Diseño Conceptual, primeramente se define los objetos y su funcionalidad –no necesariamente un diseño definitivo sino sólo para estimación. Luego se clasifican los objetos por tipo y tamaño y se busca en bases de datos históricas información sobre lo más parecido que se ha desarrollado.


Nombre Objeto	Matriz
Tipo	Datos
Métodos	13
Métodos LOC	114.9
Tamaño	Mediano



Estimación a través de Puntos de Función

La técnica de medición del tamaño en punto-función consiste en asignar una cantidad de “puntos” a una aplicación informática según la complejidad de los datos que maneja y de los procesos que realiza sobre ellos, siempre tratando de considerarlo desde el punto de vista del usuario.

Los puntos de función se calculan completando la tabla siguiente:

Parámetros de medición	Factor de ponderación					Cuenta
	Cuenta	Simple	Medio	Complejo		
Número de entradas de usuario	<input type="text"/>	× 3	4	6	=	<input type="text"/>
Número de salidas de usuario	<input type="text"/>	× 4	5	7	=	<input type="text"/>
Número de peticiones de usuario	<input type="text"/>	× 3	4	6	=	<input type="text"/>
Número de archivos	<input type="text"/>	× 7	10	15	=	<input type="text"/>
Número de interfaces externas	<input type="text"/>	× 5	7	10	=	<input type="text"/>
Cuenta total						<input type="text"/>



Se determinan cinco características de dominios de información y se proporcionan las cuentas en la posición apropiada de la tabla. Los valores de los dominios de información se definen de la forma siguiente:

Número de Entradas de Usuario: Se cuenta cada entrada de usuario que proporciona diferentes datos orientados a la aplicación. Las entradas se deberían diferenciar de las peticiones, las cuales se cuentan de forma separada.

Número de Salidas de Usuario: Se cuenta cada salida que proporciona al usuario información orientada a la aplicación. En este contexto la salida se refiere a informes, pantallas, mensajes de error, etc. Los elementos de datos particulares dentro de un informe no se cuentan de forma separada.

Número de Peticiones de Usuario: Una petición se define como una entrada interactiva que produce la generación de alguna respuesta del software inmediata en forma de salida interactiva. Se cuenta cada petición por separado.

Número de archivos: Se cuenta cada archivo maestro lógico (esto es, un grupo lógico de datos que puede ser una parte de una gran base de datos o un archivo independiente).

Número de interfaces externas: Se cuentan todas las interfaces legibles por la máquina (por ejemplo: archivos de datos de cinta o disco) que se utilizan para transmitir información a otro sistema.

Esta cuenta total nos da como resultado los Puntos de Función Brutos o Puntos de Función sin Ajustar.

Una vez que se han recopilado los datos anteriores, a la cuenta total o puntos de función brutos se asocia un valor de complejidad. Se calcula el Factor de Ajuste para obtener los Puntos de Función Ajustados.



No.	Pregunta
1.	¿Requiere el sistema copias de seguridad y de recuperación fiables?
2.	¿Se requiere comunicación de datos?
3.	¿Existen funciones de procesamiento distribuido?
4.	¿Es crítico el rendimiento?
5.	¿Se ejecutaría el sistema en un entorno operativo existente y fuertemente utilizado?
6.	¿Requiere el sistema entrada de datos interactiva?
7.	¿Requiere la entrada de datos interactiva que las transacciones de entrada se lleven a cabo sobre múltiples pantallas u operaciones?
8.	¿Se actualizan los archivos maestros de forma interactiva?
9.	¿Son complejas las entradas, las salidas, los archivos o las peticiones?
10.	¿Es complejo el procesamiento interno?
11.	¿Se ha diseñado el código para ser reutilizable?
12.	¿Están incluidas en el diseño la conversión y la instalación?
13.	¿Se ha diseñado el sistema para soportar múltiples instalaciones en diferentes organizaciones?
14.	¿Se ha diseñado la aplicación para facilitar los cambios y para ser fácilmente utilizada por el usuario?

Cada una de las preguntas anteriores es respondida usando una escala con rangos desde 0 (no importante o aplicable) hasta 5 (absolutamente esencial), y luego se suma el total de los pesos ponderados para determinar el valor de ajuste de complejidad total y utilizar este dato en el cálculo del factor de ajuste. La escala es la siguiente:

0	1	2	3	4	5
Sin influencia	Incidental	Moderado	Medio	Significativo	Esencial



Para calcular el factor de Ajuste utilizamos la siguiente fórmula:

$$FA = [0,65 + 0,01 \times \Sigma (Fi)]$$

Fi donde i puede ser de uno hasta 14 los valores de ajuste de complejidad basados en las respuestas a las siguientes preguntas:

Una vez que se calcularon los Puntos de Función Brutos y el Factor de Ajuste se procede a calcular los Puntos de Función Ajustados, utilizando la relación siguiente:

$$FP = \text{cuenta-total} \times [0,65 + 0,01 \times \Sigma (Fi)]$$

Donde cuenta-total es la suma de todas las entradas obtenidas o Puntos de Función Brutos.

$$FPA = FPB \times FA$$

Finalmente se aplican las Fórmulas para estimar la Productividad, Esfuerzo, Costo Total y Costo por Punto de Función, utilizando las fórmulas siguientes:

$$\text{Productividad} = \text{Tiempo} / \text{Puntos de Función}$$

$$\text{Esfuerzo} = \text{Productividad} * \text{Tamaño del Software}$$

$$\text{Costo Total} = \text{Tamaño de la Aplicación} * \text{Costo}$$

$$\text{Costo Unitario} = \text{Costo Total} / \text{FP}$$



A continuación se aplica el método de Estimación por Puntos de Función al Sistema Automatizado de Control de Bienes, -INETER.

1. Calculamos los Puntos de Función Brutos usando la tabla de Factores de Ponderación:

Dominios de Información	Valores	Ponderación Media	Total
Entradas	30	4	120
Salidas	33	5	165
Consultas	60	4	240
Archivos	37	10	370
Interfaces	0	7	0
Puntos de Función			895

Puntos de Función Brutos = 895 FP

2. Calculamos el Factor de Ajuste usando los pesos de las respuestas a las 14 preguntas de estimación de complejidad del sistema.

Preguntas	Rango
1. Copias de seguridad y de recuperación fiables	5
2. Comunicación de datos	5
3. Funciones de procesamiento distribuido	0
4. Rendimiento crítico	5
5. Entorno operativo y fuertemente utilizado	5
6. Entrada de datos interactiva	5
7. Transacciones sobre múltiples pantallas	0
8. Actualización interactiva de archivos maestros	5
9. Entradas, salidas, archivos o peticiones complejas	5
10. Procesamiento interno complejo	4
11. Código reutilizable	1
12. Conversión e instalación	0
13. Múltiples instalaciones en diferentes organizaciones	0
14. Facilitar cambios y ser fácilmente utilizada	5
Nivel de Influencia	45



Aplicamos la fórmula del Factor de Ajuste:

$$FA = [0,65 + 0,01 \times \Sigma (Fi)]$$

$$FA = [0,65 + 0,01 \times 45]$$

$$FA = 1.1$$

Factor de Ajuste = 1.1

3. Estimamos los Puntos de Función Ajustados que son el resultado de multiplicar los Puntos de Función Brutos por el Factor de Ajuste.

$$FPA = FPB \times FA$$

$$FPA = 895 \text{ FP} \times 1.1$$

$$FPA = 984.5 \text{ FP}$$

$$FPA = 985 \text{ FP}$$

Puntos de Función Ajustados = 985 FPA

Luego calculamos la Productividad, Esfuerzo, Costo Total y Costo por Punto de Función

$$\text{Productividad} = \text{Tiemno} / \text{Puntos de Función}$$

Según experiencias en proyectos anteriores, se estima un tiempo de 10 meses para desarrollar el sistema con un grupo de trabajo compuesto por 5 programadores.

Convertimos los meses a horas laboradas

$$10 \text{ meses de trabajo} \times 20 \text{ días laborables} \times 8 \text{ horas de trabajo} = 1600 \text{ horas}$$

$$\text{Productividad} = 1600 \text{ horas} / 985 \text{ FPA}$$

Productividad = 1.62 horas/FPA



$$\text{Esfuerzo} = \text{Productividad} * \text{Tamaño del Software}$$

$$\text{Esfuerzo} = 1.62 \text{ horas /PFA} * 985 \text{ PFA}$$

$$\text{Esfuerzo} = 1596.7 \text{ horas}$$

$$\text{Esfuerzo} = 1597 \text{ horas}$$

$$\text{Costo Total} = \text{Tamaño de la Aplicación} * \text{Costo Unitario}$$

Por experiencias en proyectos anteriores sobre duraciones de tiempo y estimaciones basadas en costos actuales de recursos recopiladas por el equipo informático de la organización se estima un costo de U\$ 160 000 para el desarrollo de este sistema. Siendo así:

$$\text{Costo Total} = \text{U\$ } 160,000$$

$$\text{Costo Unitario} = \text{Costo Total} / \text{FP}$$

$$\text{Costo Unitario} = \text{U\$ } 160,000 / 985 \text{ PFA}$$

$$\text{Costo Unitario} = 162.44 \text{ U\$/PFA}$$

$$\text{Costo Unitario por Punto de Función} = \text{U\$ } 163$$

Los resultados indican que la productividad de cada programador es de 1.62 horas/PFA, es decir que para cada unidad de software que en este caso es el punto de función ajustado, un programador se tarda 1.62 horas en realizarlo, en caso de que se desee aumentar la productividad se deben aumentar las horas de trabajo o bien bajar la complejidad del sistema a desarrollar. El esfuerzo indica que se necesitan 1597 horas para desarrollar el proyecto. El costo total está determinado mediante la realización de presupuesto en base a experiencias históricas del cuerpo informático de INETER, y el costo unitario indica que cada unidad de software cuesta U\$ 163 en ser desarrollada.



En la actualidad los modelos de estimación de costos para proyectos de desarrollo de software sirven, pero tienen muchas limitaciones. Es muy difícil predecir los costos de proyectos de desarrollo de software y en general los presupuestos que se desarrollan no son precisos.

Los patrones de gasto varían mucho entre empresas y lo ideal es desarrollar un modelo de estimación de costo que se adapte al contexto de trabajo de la empresa en que se va a desarrollar. Es muy importante que se tomen en cuenta los datos históricos sobre proyectos de desarrollo anteriores con respecto a fechas de entrega, costos, complejidad de los sistemas, motivos de retraso, razones de aumento en los costos, etc., para que sean tomados en cuenta en la planeación de proyectos futuros.

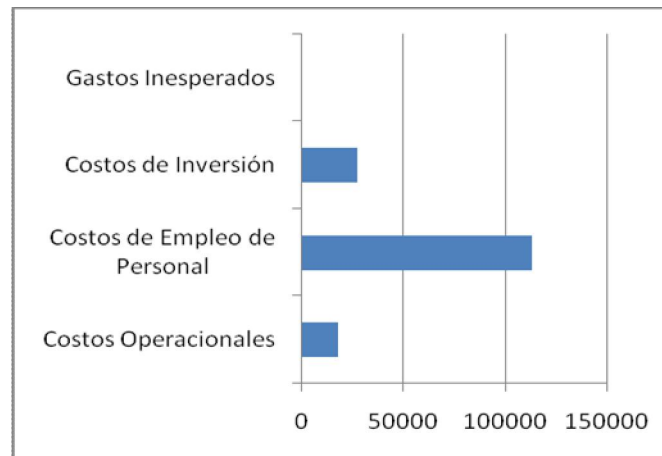
A continuación se presenta una estimación de costos para el desarrollo del Sistema Automatizado de Control de Bienes, -INETER.

Los costos presupuestados en la tabla siguiente son estimaciones, y como todo presupuesto está sujeto a variaciones en el tiempo.

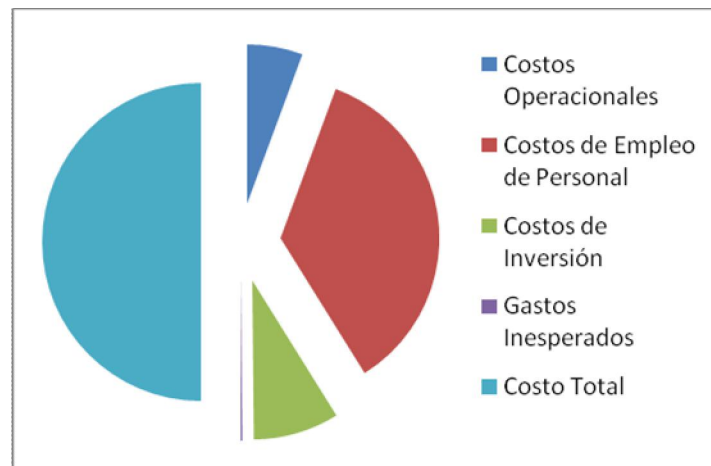


	Duración del Proyecto en meses										Total	
	1	2	3	4	5	6	7	8	9	10		
Costos Operacionales												
Papelería y Consumibles de Oficina	500	500	500	500	500	500	500	500	500	500	500	5000
Transporte	400	400	400	400	400	400	400	400	400	400	400	4000
Pago de Eventualidades	60	60	60	60	60	60	60	60	60	60	60	9000
												18000
Costos de Empleo de Personal												
Analistas de Sistemas	2000	2000	2000	2000	2000	2000	2000	2000	2000	2000	2000	20000
Programadores	7500	7500	7500	7500	7500	7500	7500	7500	7500	7500	7500	75000
Testeadores	1800	1800	1800	1800	1800	1800	1800	1800	1800	1800	1800	18000
												113000
Costos de Inversión												
Software	9889											
Hardware	17169											
Mobiliario y Equipo de Oficina	720											
												27778
Gastos Inesperados												
Provisión para Gastos Inesperados	50	50	50	50	50	50	50	50	50	50	50	500
												500
Costo Total												159278

Podemos observar que el elemento de costo más elevado es el de Empleo de Personal.



Es decir que de todos los fondos destinados al proyecto, la mayor parte será destinada al pago de los desarrolladores del software, como se muestra a continuación:



Es muy importante notar que los costos de desarrollo de software son muy elevados y que la INETER con su capacidad de pago no puede costearlos. Esto debe alertarnos a buscar formas de desarrollo de software que nos permitan bajar los costos sin afectar la calidad de los productos software terminados. La capacitación de los programadores, migración a software libre y la firme voluntad de mejorar pueden cambiar de manera positiva la situación actual del desarrollo de Software en el Instituto Nicaragüense de Estudios Territoriales, -INETER.



Impacto Técnico-Económico y Social

El Instituto Nicaragüense de Estudios Territoriales, -INETER tiene un alcance tanto nacional como internacional. Dentro de sus muchas y muy delicadas responsabilidades se encuentran la generación, resguardo, protección y procesamiento de datos Cartográficos, Catastrales, Meteorológicos, Hidrológicos, Geológicos y otros, para luego ponerlos a disposición de toda la sociedad.

También, tiene dentro de sus funciones la realización y análisis de estudios e investigaciones del medio físico que contribuyan al desarrollo socioeconómico y a la disminución de la vulnerabilidad ante desastres naturales y fenómenos naturales peligrosos. Los Bienes de INETER no pueden ser descuidados porque son las herramientas de trabajo del personal técnico, científico y administrativo que labora para la institución.

Los activos fijos no pueden ser descuidados en ningún momento porque a través de la utilización eficiente de estos, se puede llevar a la institución hacia el cumplimiento de las metas organizacionales de contribuir a la soberanía alimentaria de toda la población, descubrir fuentes potenciales de energía renovable y prevenir desastres naturales para proteger las vidas y los bienes de la población.

La relevancia de las funciones de INETER para los sectores económicos y sociales de Nicaragua hacen que todas las mejoras que puedan darse dentro de la institución tengan consecuencias positivas para el desarrollo de Nicaragua. Al final de cuentas, las mejoras para INETER son mejoras para Nicaragua y los nicaragüenses.



III. CONCLUSIONES

A través de la investigación para conocer las causas de la problemática del Departamento de Control de Bienes se descubrió que los procesos de dicho departamento presentan la necesidad de ser mejorados y automatizados.

Las ventajas que la Automatización del Sistema de Control de Bienes son, entre otras:

- Las operaciones de cálculo se realizan más rápido por computadora que mentalmente.
- Los cálculos realizados por computadora tienen más exactitud y precisión que los cálculos realizados mentalmente.
- En un sistema automatizado debidamente diseñado los datos se ingresan una sola vez y se llaman cuando se necesitan, de esa forma se evita la redundancia de datos y se elimina la posibilidad de inconsistencias en los mismos datos.
- La manipulación malintencionada de datos puede ser combatida mediante medidas de seguridad de acceso al sistema y a los datos. Solamente las personas autorizadas pueden acceder al sistema y a los datos.
- La búsqueda, selección y clasificación de información es más rápida por computadora que manualmente.

El Análisis, Diseño y Desarrollo del Prototipo del Sistema Automatizado de Control de Bienes en el Instituto Nicaragüense de Estudios Territoriales –INETER, presenta una herramienta que puede mejorar las condiciones de trabajo actuales y de esa forma agilizar las transacciones, aumentar la confiabilidad, seguridad en el almacenamiento y recuperación de la información, así como reducir la cantidad de errores que ocurren en las actividades diarias del Departamento de Control de Bienes. Este producto software también contribuye a satisfacer la demanda de información para la toma de decisiones en el instituto.